1963

Simulation study of some digital control methods for shipboard fire control.

Stapp, Aron L.

Monterey, California: U.S. Naval Postgraduate School

SIMULATION STUDY OF SOME DIGITAL CONTROL METHODS FOR SHIPBOARD FIRE CONTROL

ARON L. STAPP

SIMULATION STUDY
OF SOME DIGITAL CONTROL METHODS
FOR SHIPBOARD FIRE CONTROL

by

Aron L. Stapp

Lieutenant Commander, United States Navy

Submitted in partial fulfillment of
the requirements for the degree of

MASTER OF SCIENCE
IN
ENGINEERING ELECTRONICS

United States Naval Postgraduate School
Monterey, California

1 9 6 3

SIMULATION STUDY OF SOME DIGITAL CONTROL METHODS FOR SHIPBOARD FIRE CONTROL

* * * * *

Aron L. Stapp

SIMULATION STUDY

OF SOME DIGITAL CONTROL METHODS

FOR SHIPBOARD FIRE CONTROL

by

Aron L. Stapp

This work is accepted as fulfilling

the thesis requirements for the degree of

MASTER OF SCIENCE

IN

ENGINEERING ELECTRONICS

from the

United States Naval Postgraduate School

ABSTRACT

Digital control offers exciting possibilities for accuracy and flexibility that cannot be achieved by analog means. The USQ-20 General Purpose Digital Computer is on board several ships today for the Navy Tactical Data System, and an effort is made to show how such a computer might also be employed directly in closed-loop shipboard fire control systems. Some of the theory of sampled data systems and digital compensation is discussed. A digital computer program which simulates the action of a sampled data system with a digital controller is employed to experimentally determine the behaviour of various physical plants subject to digital control. An evaluation is made of the noise rejection capabilities, steady state error, and transient control of a digital process. A full description and instructions for its use is given for the simulator program, which has proved to be very helpful in making theoretical studies of control system behaviour.

The writer wishes to express his appreciation for the assistance and encouragement given him by Professor Mitchell L. Cotton of the U. S. Naval Postgraduate School and John B. Slaughter of the U. S. Navy Electronics Laboratory in this investigation.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Continued)

LIST OF ILLUSTRATIONS (Continued)

1. Introduction.

A control system which contains a digital computer may be classed as a sampled-data system. Such systems are becoming increasingly more common, both in military operations and in the commercial field. By the use of sampling in control systems it is possible to build simple, sensitive, and efficient power control devices. Sampling permits the control of tremendous power by sensitive control elements without excessive amplification, and also minimizes the loading effect upon sensitive instruments. Utilizing sampled data and digital components in a control system allows time sharing of important parts of the system. This is a very significant advantage of digital control systems.[/1/]

Since sampled-data can readily be coded, the data signals in digital control systems are received and transmitted in pulse-code form, and they will provide almost error-free channels for transmission through noisy media. Consequently, the only noise in the transmission of pulse-coded signals is the quantization error. Likewise, a digital computer in a control system allows data processing of control information so that the flexibility and versatility of the digital computer can be utilized to improve the performance of the control system. Furthermore, digital control techniques make feasible the system compensation by non-linear programming and adaptive, or self-optimizing, control. This is another significant advantage achievable

with digital control.[2]

The principal features of a typical system containing a digital computer are shown in block diagram form in Figure 1.

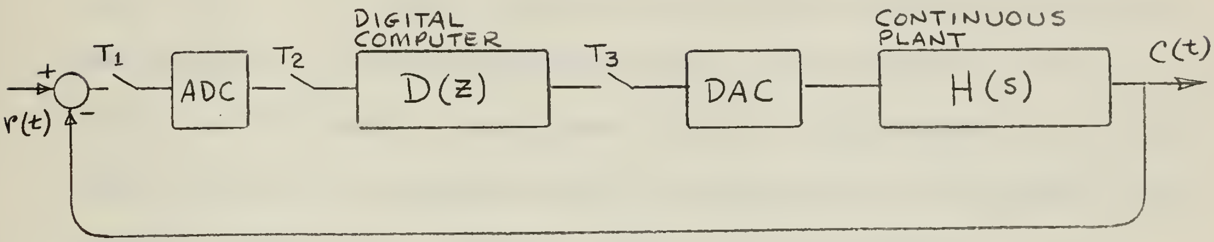


Figure 1. Typical block diagram of a hybrid control system.

A characteristic feature of this system is the fact that data will appear at one or more places within it as a sequence of numbers or fixed signal levels that do not change appreciably except at fixed instants in time. The process that converts continuous data into such a sequence is called the sampling process, and the mechanism that implements the process is called an analog-to-digital converter (ADC). The process that converts the output number sequence of the digital computer to a more or less continuous analog quantity is called the desampling process, and the mechanism accomplishing this process is called the digital-to-analog converter (DAC).[2]

The quantities T_1, T_2, and T_3 are the periods of the samplers shown. In almost every case T_1 is the same as T_2, but the relation between T_2 and T_3 may be chosen by the designer. The simplest relation, of course, is when T_2 equals T_3. The system is then called <u>single-rate</u> since all discrete

number sequences appearing within the system have the same sampling period. A multi-rate system occurs when T_2 and T_3 are not the same. Analysis is simplified when T_2 and T_3 are related by an integer and we have "synchronous" sampling. It is also simpler to analyze a multi-rate system wherein T_3 is less than T_2 (ie., $T_3 = T_2/n \quad n = 2, 3, \ldots$) which is fortunate, since most of the advantages of a multi-rate system are derived when the output of the digital controller occurs at a faster rate than the input.

The experimental results described in this thesis are obtained from a digital computer program that simulates a hybrid control system from beginning to end. See Appendix I for the description of this program. In an actual control system the only operations that would perforce be external to the digital computer would be the plant itself and the ADC. In the investigation conducted by John Slaughter at the U. S. Navy Electronics Laboratory, San Diego, California, a Packard-Bell Model M2 analog-to-digital converter is employed, and a 100 micro-second conversion and settling time is allowed. For a discussion of a proposed analog-to-digital system for use with the CDC 1604 digital computer see Reference 6.

The generation of the error signal by comparison between input and output signals can be accomplished either externally to the digital computer or by the digital computer itself. The latter case would require a two channel ADC and a multiplexing system for sampling respectively the

input and the output signals.

Data reconstruction by the DAC can also be performed by an external device and can be of various degrees of complexity. However, the fact that the plant in control systems is usually low-pass in frequency response makes the use of overly complex data-reconstruction systems unnecessary. Generally speaking, the zero-order data hold is found to be adequate for most systems found in practice. This being so, the data reconstruction process can easily be implemented by the digital computer since an output register of the computer can be utilized as a zero-order hold.

2. The effect of sampling.

Sampling is the act of examining a continuous function at specified increments of an independent variable, usually time. Sampling processes can and do assume a wide variety of forms and can be separated roughly into three categories:

1. Linear, fixed pattern sampling
2. Non-linear, signal dependent sampling
3. Random sampling

Included in the first category is the historically conventional concept of "instantaneous" sampling of fixed period T. By instantaneous it is meant that the duration, or pulse width, of the sample of the signal is very short compared to the shortest time constant of the system. Under this assumption (which will be employed in this thesis) the sampling process can be regarded as generating a sequence of impulses. This viewpoint leads to the development of the Z-transform calculus and also to the use of difference equations as a mathematical tool. Extensive tables of Z-transforms may be found in the literature, and it is relatively easy to take the Z-transform of simple, low-order transfer functions. However, for a fifth order plant or higher, finding the Z-transform is quite laborious. Appendix 2 describes a digital computer program written for transfer functions of Type 0 and Type 1 servo-mechanisms up to the tenth order.

The Sampling Theorem by Shannon states that a band-limited signal of highest frequency F_o can be completely

specified by sampling at a rate of $2F_o$. Recovery of the signal at this minimum theoretical sampling rate requires an ideal filter. In practice, signals are not ordinarily band-limited and filters do not have ideal attenuation characteristics. Even at high sampling rates, sampled-data systems usually exhibit distortion in the form of "ripple". One means of combating this ripple is to employ a multi-rate digital controller whose output occurs at a faster rate than the basic error-sampling rate of the over-all system. By applying control signals to the plant more frequently much of the ripple can be reduced and the plant can be made to respond more quickly to an input signal. Design judgment must be used, however, since the output signals from a multi-rate controller are frequently of greater magnitude than those from a single-rate device, and the possibility of plant saturation as well as the added complexity of the controller must be balanced against the advantages to be gained from a multi-rate system.

Of crucial importance in a sampled-data system is the effect of sampling on the stability of the system. In an otherwise stable system, sampling reduces the degree of stability and can even lead to instability./2/ There are various criteria for continuous systems that can be applied to sampled-data systems. Among these are the modified Routh-Hurwitz criterion, the Schur-Cohn criterion, and the Nyquist criterion. As an example of the determination of stability using the modified Routh-Hurwitz criterion we will use the third order Type 1 servo-mechanism shown in

Figure 2. The plant is closed-loop stable with the gain indicated. We desire to convert the continuous system to a sampled-data system, so we first insert sampling and data reconstruction as shown. The matter of digital compensation will be held in abeyance while we investigate the effect of sampling only.

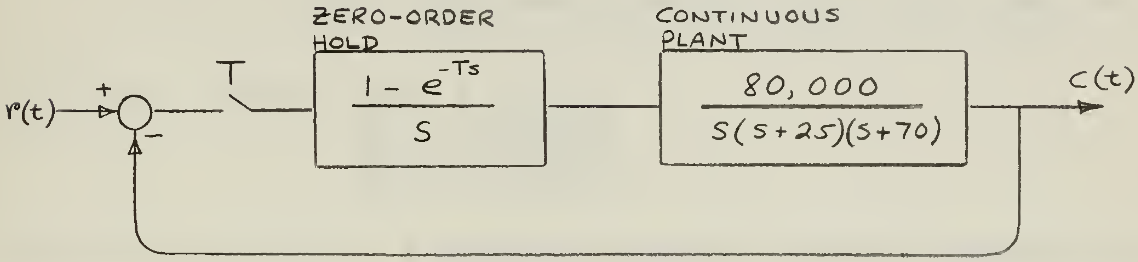


Figure 2. A third order sampled-data system.

Suppose that a sampling rate of 100 samples per second, or T = .01 seconds, has been chosen a priori. Then the modified Routh-Hurwitz procedure gives a "yes or no" result in regard to stability. For this value of T the forward path pulse transfer function (or Z-transform) is:

$$G(z) = \frac{P(z)}{Q(z)} = \frac{.010600\,z^2 + .033672\,z + .006584}{z^3 - 2.275386\,z^2 + 1.662127\,z - 0.386741}$$

For the unity feedback configuration shown the closed loop transfer function has a characteristic equation given by:

$P(z) + Q(z) = A(z) = A3\ z^3 + A2\ z^2 + A1\ z + A0$

where
A3 = 1.000000
A2 = -2.264786
A1 = 1.695799
A0 = -0.380157

The closed loop sampled-data system is stable if the roots of A(z) lie within the unit circle in the Z-plane. By

employing the Bilinear Transformation, $Z = \frac{1 + w}{1 - w}$, the interior of the Z-plane is mapped into the left half of the w-plane. Once the modified characteristics equation is obtained, namely: $A(w) = B3\ w^3 + B2\ w^2 + B1\ w + B0$ then the Routh array may be formed and the criterion for a third order system may be applied. This criterion is:

$B1 \cdot B2 - B0 \cdot B3$ is a positive quantity if all the roots of $A(w)$ are in the left half plane

In this example, B3 = 5.340742
B2 = 2.428516
B1 = 0.179886
B0 = 0.050856

so $B1 \cdot B2 - B0 \cdot B3 = + 0.165247$ and the system is stable for T equals .01 seconds.

If it is desired to decrease the sampling rate as much as possible in order to time-share the computer or to allow for increased computation time between samples, then it becomes necessary to know the limit of stability as the sampling period T is increased. Unfortunately, the characteristic equation is transcendental in T, and we cannot solve directly for T at the stability limit. Iterative procedures most readily delegated to a digital computer must be employed. Appendix 3 shows the development of the characteristic equation and the Fortran program for the CDC 1604 digital computer which calculates the Routh-Hurwitz criterion for this third order system for one hundred values of T ranging from .001 seconds to 0.1 seconds. From the results shown the stability limit on T is evidently near .025 seconds, since this is the last value for which the R-H criterion is a positive quantity.

Figure 3 shows the response of this un-compensated third order sampled data system to a unit step input for three values of T. For T of .001 seconds the response is indistinguishable to the eye from the results obtained in an analog computer simulation of the problem. This would be a good value to choose for the time increment in a solution of the plant differential equation by numerical methods (see Appendix 1). It can be seen that the stability of the system is degraded somewhat when T is increased to .01 seconds, but the system is still stable. This is the sampling period to be employed later when we introduce digital compensation to this third order system. A sampling period of .05 seconds results in severe instability, as we could expect from the Routh-Hurwitz determination. Digital compensation is a <u>must</u> in this case, since stability is a prime requisite. Whether or not we compensate a stable system depends on how satisfactory the response is without compensation.

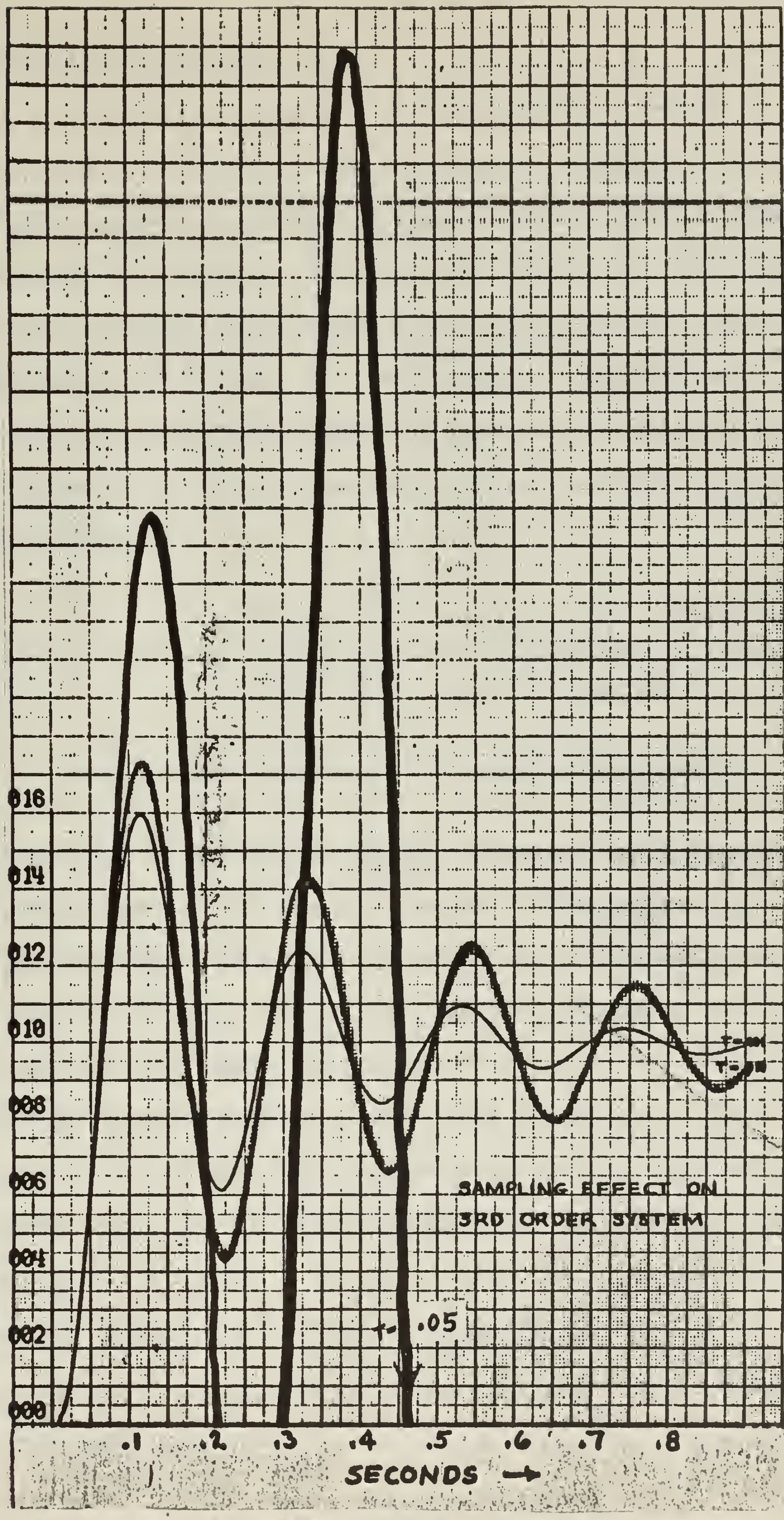


Figure 3. Sampling effect on a third order system

3. The digital process.

As previously mentioned, a sampled-data system is characterized by the fact that in certain areas of the system signals appear as discrete quantities or fixed numbers in a sequence at regular intervals of time. The most obvious point at which this occurs is at the digital computer or controller. The input to the computer is a sequence of numbers and the computer output is also a sequence of numbers. We can thus regard the computer as a "black box" as shown in Figure 4.

Figure 4. Generalized concept of a digital computer.

The input sequence, $E1\left[pT_1\right]$, consists of N discrete values separated in time by T_1 seconds. The output sequence may contain a different number, M, of fixed quantities and the period, T_2, of the output sequence need not be the same as the input period.

For simplicity let us first take T_1 equal to T_2. Then at a certain time, pT, the general form of linear computation that the digital computer can perform may be expressed as:

$$e2[pT] + b1 \cdot e2[(p-1)T] + \cdots + b_{M-1} \cdot e2\left[(p-M+1)T\right]$$

$$= a0 \cdot e1[pT] + a1 \cdot e1\left[(p-1)T\right] + \cdots + a_{N-1} \cdot e1\left[(p-N+1)T\right]$$

This can be written: $$e2[pT] = \sum_{K=0}^{N-1} a_K e1_{p-K} - \sum_{j=1}^{M-1} b_j e2_{p-j}$$

or, by Z-transform calculus: $$\frac{E2(z)}{E1(z)} = \frac{\sum a_K z^{-K}}{1 + \sum b_j z^{-j}}$$

which can be considered the pulse transfer function of a digital computer. If all the b_j are zero then we have what is called a <u>finite memory</u> process since the output at present time, pT, is the weighted sum of the present input plus earlier inputs up to some finite limit, N. If not all of the b coefficients are zero then a weighted contribution of earlier outputs is also involved in forming the output at time pT. Since each earlier output was a combination of both input and output sequences, <u>ad infinitum</u>, we now have what is called an infinite memory process. Another way of showing this is to perform long division on the transfer function to obtain new weighting coefficients, c_k, for the input sequence alone. Only in the case where $a_k Z^{-k}$ contains $1 - b_j Z^{-j}$ as a factor does long division fail to result in an infinite series, showing the need for an infinite memory of previous inputs. When such a transfer function describes the over-all closed loop process, we have what is known as a "non-finite settling time" process and the systematic error (or difference between input and output) can never theoretically reach zero.

In the system configuration employed in this thesis we will use the concept of the digital computer transfer

function in two ways. First is the over-all pulse transfer function of the closed-loop process, defined as:

$$K(z) = \frac{C(z)}{R(z)} = \frac{\text{OUTPUT SEQUENCE}}{\text{INPUT SEQUENCE}}$$

Our attention will be devoted mostly to finite memory K(z) consisting of numerator polynomial only. Such a process has a finite settling time and can "forget" the effect of initial errors or acquisition transients in the input signal.

Second is the pulse transfer function of the digital controller in the forward path. Storage will be allocated for a finite number of values in both the input error sequence and the output error sequence, and the controller transfer function will be a ratio of polynomials in Z^{-1} defined as: $D(z) = \frac{Eout(z)}{Ein(z)} = \frac{\text{processed error sequence}}{\text{observed error sequence}}$

In case the input sequence to the controller has a period, T_1, different from the period, T_2, of the output sequence we can still write a pulse transfer function that is amenable to interpretation using difference equations providing T_1 and T_2 are related by an integer. The pulse transfer function is then expressed in the "Z-domain" created by the faster sampler, which in this thesis will be at the output from the controller. In the next section is provided the mathematical basis for these assertions.

4. Response criteria.

In view of the flexibility possible with active digital compensation there is a very large number of possible over-all response functions which can be implemented. As a starting point, the simplest or "minimal" prototype response functions are convenient. Minimal prototype systems are approached with the point of view that they must be able to respond satisfactorily to some convenient test input such as a step, ramp, or constant acceleration, or all three. The requirements which are set for minimal prototype response functions are:

a. The over-all response and the response of all elements of the system must be physically realizable.

b. The steady-state response to the test input must have zero systematic error.

c. The transient response should be as fast as possible and the settling time should be equal to a finite number of sampling intervals./4/

The following table shows the test inputs and the required minimal prototype response functions, K(z):

Input	K(z)
Step	z^{-1}
Ramp	$2z^{-1} - z^{-2}$
Acceleration	$3z^{-1} - 3z^{-2} + z^{-3}$

These minimal responses are possible only for those plants whose pulse transfer functions, G(z), contain no zeros or poles outside the unit circle in the Z-plane.

The essence of digital compensation is that the constant coefficients of the pulse transfer function, D(z), for the controller are adjusted so that the desired over-all pulse transfer function of the process, K(z), is realized. Depending on the structure of the sampled-data feed-back system, the controller transfer function is derived from system functions in various ways. The configuration employed in this thesis is the typical one shown earlier in Figure 1, viz., in-line digital compensation of a unity feed-back, error-sampled control system. The ADC can be thought of as a single sampling switch operating with period T. Data reconstruction will be accomplished by a zero-order hold, and the pulse transfer function of this DAC in combination with the continuous plant will be denoted by G(z). Since the actual input and output are continuous signals, fictitious samplers are shown producing R(z) and C(z) so that we may speak of the over-all pulse transfer function, K(z). For convenience, Figure 1 is re-drawn embodying these ideas as Figure 5 below.

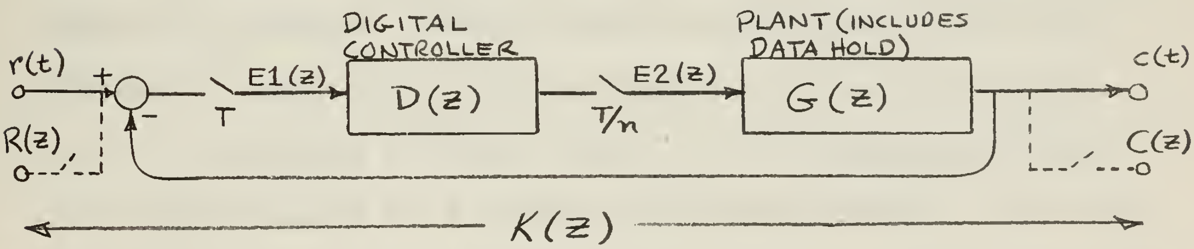


Figure 5. Sampled-data system with in-line compensation.

Taking the single-rate case first (where n=1, thus T/n=T), the over-all pulse transfer function is seen to be

$$K(z) = \frac{C(z)}{R(z)} = \frac{D(z)\cdot G(z)}{1 + D(z)\cdot G(z)}$$

which can be solved to give

$$D(z) = \frac{1}{G(z)} \cdot \frac{K(z)}{1 - K(z)}$$

Depending on the poles and zeros of the plant which it is desired to compensate digitally, complete freedom of choice of K(z) may not exist. According to Franklin and Ragazzini (4) the restrictions on K(z) are:

> It is necessary that the specified over-all pulse transfer function, K(z), contain as its zeros all those zeros of the plant pulse transfer function, G(z) which lie on or outside the unit circle in the z-plane, and that 1-K(z) contain as its zeros all those poles of the plant transfer function which lie on or outside the unit circle in the z-plane.

Systems which are designed for minimum and finite settling time often do not give good performance when subjected to an input other than that for which they are designed. In this sense, minimal systems may be regarded as highly "tuned". In addition, the severe shocks which result in the plant cause substantial ripple in the continuous output even though the output is error-free at the sampling instants. These effects were noted in the literature[7] and led to the introduction of a term in the over-all response function known as the <u>staleness</u> factor. This factor leads to a softening of the response, with the result that the system can be expected to respond slightly slower but adequately to a number of test inputs. The choice of staleness factor can be arrived at by optimizing procedures[8] or by observing the response to some most

likely form of test input. The staleness factor is introduced as the constant "c" in the following expression for the over-all prototype transfer function:

$$K(z) = \frac{a \cdot K(z)\text{minimal}}{(1-cz^{-1})^N}$$

where in practice, $N = 1$, c is a positive constant less than unity, and "a" is determined by applying the Final Value Theorem.

In order to compensate the system so that the output is ripple-free after a finite settling time the following design rules must be applied:/4/

a. All the rules for minimal prototype response systems apply.

b. The open loop transfer function must be capable of generating an output that is the same as the input.

c. The over-all pulse transfer function, K(z), must contain as its zeros <u>all</u> the zeros of the plant pulse transfer function, G(z), and not just the zeros of G(z) which lie outside the unit circle in the z-plane.

Let us turn now to the case in which the output of the digital controller occurs more frequently than the input. The development that follows is due to Kranc (Reference 9).

If we apply a discrete sequence, E(z), to a continuous plant, H(s), then it follows that the continuous output is:

$$C(s) = H(s) \cdot E(z) \tag{1}$$

For <u>n</u> equal unity we may take the pulse transform of both sides of equation (1) for a sample period of T. Since E(z) is already a function of Z it is invariant under the trans-

formation, so the result is: $C(z) = H(z) \circ E(z)$

When the output is sampled "n" times more frequently than the input ($n = 2, 3, \ldots$) let us adopt the notation that Z_n denotes a pulse transform taken for a sampling period of T/n. Then we may define:

$$C(Z_n) \triangleq \mathcal{Z}_n\{C(s)\} = \mathcal{Z}_n\{H(s)\ E(z)\}$$

$$H(Z_n) \triangleq \mathcal{Z}_n\{H(s)\}$$

Once again $E(z)$ is invariant (also see Reference 4, page 88) so $C(Z_n) = H(Z_n) \cdot E(z)$ but we have a mixture of z and Z_n transforms, and the meaning is unclear. If we now define

$$z_n \triangleq z^{1/n} \triangleq e^{sT/n}$$

then it can be seen that $Z = Z_n^n$ and $E(Z) = E(Z_n^n)$ so in the Z_n "domain" we can write $C(Z_n) = H(Z_n)\ E(Z_n^n)$. (2)

Returning now to the closed loop system of Figure 5 we can see the following relationships:

$$K(z) = \frac{C(z)}{R(z)} \quad \text{and} \quad K(z_n) = \frac{C(z_n)}{R(z_n)}$$

$$E1(z) = R(z) - C(z) \tag{3}$$

$$C(z) = E1(z)\ \mathcal{Z}\{D(z_n) \cdot G(z_n)\} \tag{4}$$

$$C(z_n) = E1(z) \cdot D(z_n) \cdot G(z_n) \tag{5}$$

Using (3) and (4): $$E1(z) = \frac{R(z)}{1 + \mathcal{Z}\{D(z_n) \cdot G(z_n)\}} \tag{6}$$

Substituting (6) into (5): $$C(z_n) = \frac{R(z) \cdot D(z_n) \cdot G(z_n)}{1 + \mathcal{Z}\{D(z_n) \cdot G(z_n)\}} \tag{7}$$

Since $K(z_n) = \dfrac{C(z_n)}{R(z_n)}$ there results: $K(z_n) = \dfrac{D(z_n)\cdot G(z_n)}{1 + \mathcal{Z}_1\{D(z_n)\cdot G(z_n)\}}$

which can be re-written as:

$$K(z_n)\mathcal{Z}_1\{D(z_n)\cdot G(z_n)\} - D(z_n)\cdot G(z_n) + K(z_n) = 0 \qquad (8)$$

Equation (8) is a functional equation of the form

$$\mathcal{Z}_1\{F(z_n)\} - F(z_n) = -K(z_n)$$

It can be shown /9/ that the solution is

$$F(z_n) = \frac{K(z_n)}{1 - K(z)} = D(z_n)\cdot G(z_n)$$

so finally, the transfer function of the multi-rate controller is:

$$D(z_n) = \frac{1}{G(z_n)} \cdot \frac{K(z_n)}{1 - K(z_n^n)} \qquad (9)$$

The Z_n- transform of the plant is simply obtained by substituting Z_n for Z and T/n for T in G(z). The Z_n- transform of the overall process is not so simply obtained. We must start by imposing steady-state criteria and the restraint of physical realizability. The following development is due to Franklin and Ragazzini (Reference 4).

For inputs whose Laplace transform is $R(s) = \frac{1}{s^K}$ we can write

$$R(z_n) = \frac{P(z_n)}{(1-z_n^{-1})^K} \quad \text{and} \quad R(z_n^n) = \frac{P(z_n^n)}{(1-z_n^{-n})^K} = \frac{C(z_n)}{K(z_n)}$$

Thus the error sequence, $R(Z_n) - C(Z_n)$, can be expressed as

$$E1(z_n) = \frac{P(z_n)}{(1-z_n^{-1})^K} - \frac{P(z_n^n)K(z_n)}{(1-z_n^{-n})^K} \qquad (10)$$

For a stable process the Final Value Theorem states that

$$E1(\infty) = \lim_{z_n\to 1}(1-z_n^{-1})E1(z_n) \qquad (11)$$

In order for equation (10) to meet the requirements of equation (11) we must have

$$K(z_n) = \frac{(1-z_n^{-n})^K}{(1-z_n^{-1})^K}\cdot f(z_n) = (1 + z_n^{-1} + z_n^{-2} + \cdots + z_n^{-n+1})f(z_n) \qquad (12)$$

where $f(Z_n)$ is an arbitrary function of Z_n whose coefficients are to be determined.

Substituting (12) into (10) results in:

$$E1(Z_n) = \frac{P(Z_n) - P(Z_n^n)f(Z_n)}{(1 - Z_n^{-1})^K} \tag{13}$$

and for zero steady state error the numerator of $E1(z)$ must contain as a factor the denominator of the input, $R(Z_n)$, i.e.,

$$P(Z_n) - P(Z_n^n)f(Z_n) = (1-Z_n^{-1})^K \cdot T(Z_n) = Q(Z_n) \tag{14}$$

or equivalently:

$$\left.\frac{d^m Q(Z_n)}{d(Z_n^{-1})^m}\right|_{Z_n=1} = 0 \qquad \text{for } m=0,1,\ldots K \tag{15}$$

The restraints imposed by equation (15) are applied in order to determine the coefficients of the arbitrary function, $f(Z_n)$. Once $f(Z_n)$ is known, equation (12) gives $K(Z_n)$.

The only unknown now remaining in the fundamental design equation for the digital controller, equation (9), is $K(Z_n^n)$. This is obtained from $K(Z_n)$ by merely taking the j^{th} powered terms of $K(Z_n)$. ...$j = n, 2n, 3n, \ldots$

In very special circumstances, $K(Z_n)$ may be found in a simpler way by using so-called "single-rate" techniques described in Reference 4. These special circumstances are:

a. The plant must be open-loop stable.

b. $K(Z_n)$ is limited to a finite settling time type (i.e., numerator polynomial only).

c. The speed-up factor, "n", in the controller output must be selected so that all transients at output sampling instants are over in one period of the input sampler, T. This is equivalent to

requiring that "n" equal the number of terms in the numerator of the plant's pulse transfer function, G(z).

If these circumstances obtain, and if the design goal is to achieve ripple-free response to a step input, then for a plant transfer function of

$$G(z_n) = \frac{\sum_{i=1}^{n} a_i z_n^{-i}}{\sum_{j=0}^{K} b_j z_n^{-j}} \qquad (b_0 \neq 0)$$

it can be shown/4/ that the digital controller transfer function is given by:

$$D(z_n) = \frac{\sum_{j=0}^{K} b_j z_n^{-j}}{\sum_{i=1}^{n} a_i (1 - z_n^{-1})}$$

The main limitation of this design procedure using single-rate techniques is that the speed-up factor, "n", must be the same as the number of numerator terms in G(z), which may be out of the question for a high order plant. For the fifth order plant to be discussed later a double-rate controller gave improved performance, but this controller could not be designed by the above procedure. It is also worth noting that the number of terms in $D(Z_n)$ cannot be less than "n".

Up to this point the discussion of response criteria has been predicated upon the input being a deterministic, noise-free signal, but it also may be desirable to judge

a process on the basis of its noise rejection capabilities. The following remarks concerning noise rejection are due to Monroe (Reference 1).

In order to consider an input signal which is contaminated by noise, certain simplifying assumptions are necessary. First, assume that the system in the absence of noise would perform the ideal operation on the input, ie., the systematic error is zero. Second, assume the signal and noise are uncorrelated. Last, assume the noise is "white" noise over at least one sampling period. Then it can be shown that the mean squared error of the process is:

$$\overline{e^2(nT)} = \sigma_N^2 \sum_{K=1}^{N} a_K^2$$

where σ_N^2 is the mean squared value of the input noise

a_K is a coefficient of the digital process, K(z).

$$K(z) = \sum_{K=1}^{N} a_K z^{-K}$$

The "variance reduction factor" is defined as the ratio of the mean squared error of the output function to the mean squared value of the input noise and is seen to be the sum of squares of the coefficients of the digital process. Clearly, the smaller this factor, the better the noise rejection capabilities of the digital program will be, and in general it is desired that this factor shall be less than unity. In passing, note that the minimal prototype responses to ramp and acceleration inputs have variance reduction factors of five and nineteen, respectively.

The digital process can be constrained to have no

systematic error when the input is taken as a polynomial of a given order or lower. For example, if linear constraints are imposed, then no systematic error exists for ramp or step inputs. If, subject to linear constraints, the variance reduction factor is minimized, the following expression is obtained for the coefficients of a digital process employing N samples of past history and predicting forward α sampling intervals.

$$a_K = \frac{2(2N-1)+6\alpha}{N(N+1)} - \frac{6(N-1+2\alpha)}{N(N+1)(N-1)}K$$

and the variance reduction factor is:

$$\sum_{K=0}^{N-1} a_K^2 = \frac{2(2N-1)(N-1)+12\alpha(N-1)+12\alpha^2}{N(N^2-1)}$$

The requirements that K(z) be able to predict forward by αT arises due to the delay inherent in the computing process, D(z). The discrete error value at sampling time nT equals the input signal at time nT minus the output resulting from a control pulse applied to the plant one sample period earlier, ie., at time (n-1)T. In order to have zero error, the output must equal the input, so K(z) must predict forward one sample period, thus α = 1.

The second formula above can be solved for N, given any desired value of variance reduction factor. In the case of unity variance reduction the result shows that six samples are required, and the process pulse transfer function is:

$$K(z) = \frac{2}{3}z^{-1} + \frac{7}{15}z^{-2} + \frac{4}{15}z^{-3} + \frac{1}{15}z^{-4} - \frac{2}{15}z^{-5} - \frac{1}{3}z^{-6}$$

It was noted earlier that the presence of a plant zero on or outside the unit circle in the z-plane prevents the realization of minimal prototype response. Similarly, such a zero will prevent realization of the unity variance response. Indeed, when a zero must be accounted for, it becomes quite laborious to determine any set of coefficients, a_K, that will achieve noise reduction. According to Monroe (Reference 1) it is possible to design K(z) with noise reducing properties for a plant with a single zero outside the unit circle by the following procedure:

let b = the value of the zero to be accounted for,

α = unity for prediction forward on one sample period.

The pulse transfer function of the process using N data points is

$$K(z) = \sum_{K=1}^{N} a_K z^{-K} = (1 - bz^{-1}) \sum_{K=1}^{N-1} \beta_K z^{-K}$$

For linear constraints on the process (ie., zero systematic error for ramp or step inputs) then the matrix equation below can be solved for β_0, β_1, β_2 and β_3. These values are then employed in the recursion formula shown in order to find β_4 and so on up to β_{N-1}.

It was noted earlier that the presence of a plant zero on or outside the unit circle in the z-plane prevents the realization of minimal prototype response. Similarly, such a zero will prevent realization of the unity variance response. Indeed, when a zero must be accounted for, it becomes quite laborious to determine any set of coefficients, a_K, that will achieve noise reduction. According to Monroe (Reference 1) it is possible to design K(z) with noise reducing properties for a plant with a single zero outside the unit circle by the following procedure:

let b = the value of the zero to be accounted for,

α = unity for prediction forward on one sample period.

The pulse transfer function of the process using N data points is

$$K(z) = \sum_{K=1}^{N} a_K z^{-K} = (1 - b z^{-1}) \sum_{K=1}^{N-1} \beta_K z^{-K}$$

For linear constraints on the process (ie., zero systematic error for ramp or step inputs) then the matrix equation below can be solved for β_0, β_1, β_2 and β_3. These values are then employed in the recursion formula shown in order to find β_4 and so on up to β_{N-1}.

$$
\left[\begin{array}{cc|cccccccc}
-1 & 0 & 1+b^2 & b & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\
-1 & -1 & b & 1+b^2 & b & & & & & \vdots \\
-1 & -2 & 0 & b & 1+b^2 & b & & & 0 & \vdots \\
\vdots & -3 & \vdots & & 0 & b & 1+b^2 & & & \vdots \\
\vdots & \vdots & \vdots & & & \ddots & \ddots & \ddots & & \vdots \\
\vdots & \vdots & \vdots & & 0 & & \ddots & \ddots & \ddots & \vdots \\
\vdots & \vdots & \vdots & & & & & 1+b^2 & b & 0 \\
-1 & -(N-3) & \vdots & & & & & b & 1+b^2 & b \\
-1 & -(N-2) & 0 & \cdots & \cdots & \cdots & \cdots & 0 & b & 1+b^2 \\
\hline
0 & 0 & 1 & 1 & 1 & 1 & \cdots & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 2 & 3 & \cdots & N-4 & N-3 & N-2
\end{array}\right]
\left[\begin{array}{c}
\mu_0 \\ \mu_1 \\ \hline \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \hline \beta_{N-3} \\ \beta_{N-2}
\end{array}\right]
=
\left[\begin{array}{c}
0 \\ 0 \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ \vdots \\ 0 \\ \hline \frac{1}{1+b} \\ -\frac{b+\alpha(1+b)}{(1+b)^2}
\end{array}\right]
$$

The recursion formula is:

$$
\beta_{K+4} = -\frac{(1-b)^2}{b}\beta_{K+3} + 2\frac{(1-b+b^2)}{b}\beta_{K+2} - \frac{(1-b)^2}{b}\beta_{K+1} - \beta_K
$$

Even with a noisy input signal it is still desirable to exercise some control over the transient behaviour of the system. It is possible, again according to Monroe, to design the over-all process according to a composite criterion wherein preference can be expressed for step transient response (in the mean squared error sense) over noise rejection by means of an arbitrary parameter, β, ranging between zero and one. Subject to linear constraints and for no plant zero outside the unit circle, the process coefficients, a_K, are calculated from the following formulae:

$$a_{i+2} = \left[2 + \frac{\beta}{N(1-\beta)}\right] a_{i+1} - a_i \qquad \text{(recursion relationship)}$$

$$a_0 = \frac{\nu_1 - \nu_0}{\beta_0 \nu_1 - \beta_1 \nu_0} \qquad a_1 = \frac{\lambda_0 \nu_1 - \lambda_1 \nu_0}{\beta_0 \nu_1 - \beta_1 \nu_0} \qquad \text{(starting formulae)}$$

$$\beta_0 = \frac{\lambda_0^N - 1}{\lambda_0 - 1} \qquad \beta_1 = \frac{\lambda_1^N - 1}{\lambda_1 - 1}$$

$$\nu_0 = \frac{(N-1)\lambda_0^{N+1} - N\lambda_0^N + \lambda_0}{(\lambda_0 - 1)^2} \qquad \nu_1 = \frac{(N-1)\lambda_1^{N+1} - N\lambda_1^N + \lambda_1}{(\lambda_1 - 1)^2}$$

$$\lambda_0 = \frac{1}{2}\left\{2 + \frac{\beta}{N(1-\beta)}\right\} + \frac{1}{2}\left\{\frac{\beta}{N(1-\beta)}\right\}^{\frac{1}{2}}\left\{4 + \frac{\beta}{N(1-\beta)}\right\}^{\frac{1}{2}}$$

$$\lambda_1 = \frac{1}{2}\left\{2 + \frac{\beta}{N(1-\beta)}\right\} - \frac{1}{2}\left\{\frac{\beta}{N(1-\beta)}\right\}^{\frac{1}{2}}\left\{4 + \frac{\beta}{N(1-\beta)}\right\}^{\frac{1}{2}}$$

Figure 6 shows the Fortran coding of these equations and the results for a ten point process with beta of 0.75. The program may be used for another design merely by changing the two statements giving the values of the parameters beta and AN.

For the design of a process to a composite criterion when the plant has a zero outside the unit circle see Reference 1.

```
..JOB STAPP3
      PROGRAM STAPP3
C     COMPUTES COEFFICIENTS OF THE PROCESS K(Z) TO COMPOSITE CRITERIA
      DIMENSION AK(20)
C     B IS A PARAMETER SHOWING THE PROPORTION OF EFFORT DIRECTED AT
C     INTEGRAL SQUARED ERROR TRANSIENT RESPONSE BEHAVIOUR
      B=0.75
C     AN = THE NUMBER OF DATA POINTS TO BE USED IN THE PROCESS K(Z)
      AN=10.0
      C=B/(AN*(1.0-B))
      SQRT=0.5*(4.0*C+C*C)**0.5
      AL0=0.5*(2.0+C)+SQRT
      AL1=0.5*(2.0+C)-SQRT
      B0=(AL0**AN-1.0)/(AL0-1.0)
      B1=(AL1**AN-1.0)/(AL1-1.0)
      V0=((AN-1.0)*AL0**(AN+1.0)-AN*AL0**AN+AL0)/(AL0-1.0)**2
      V1=((AN-1.0)*AL1**(AN+1.0)-AN*AL1**AN+AL1)/(AL1-1.0)**2
      AK(1)= (V1-V0)/(B0*V1-B1*V0)
      AK(2) = (AL0*V1 - AL1*V0)/(B0*V1-B1*V0)
      NK = AN
      PRINT 8,NK,B
    8 FORMAT(15H NO. DATA PTS.=, I2, 3H B=, F3.2)
      PRINT 9
    9 FORMAT(31H COEFFICIENTS, AK(I), OF PROCESS)
      DO 10 I = 1,NK
      AK(I+2) = (2.0+C)*AK(I+1)-AK(I)
   10 PRINT 20, AK(I)
      J = NK+1
      DO 15 I=J,20
      AK(I) = 0.0
   15 PRINT 20 , AK(I)
   20 FORMAT(F20.8)
      STOP
      END
      END
```

****RESULTS*****

```
 NO. DATA PTS.=10 B=.75
 COEFFICIENTS,AK(I), OF PROCESS
          .50884589
          .29542140
          .17062333
          .09701226
          .05250487
          .02374894
          .00211768
         -.01887826
         -.04553769
         -.08585842
          .00000000
          .00000000
          .00000000
          .00000000
          .00000000
          .00000000
          .00000000
          .00000000
          .00000000
          .00000000
```

Figure 6. Program for computing composite coefficients

5. Program requirements.

Before discussing the realization of the digital compensator, D(z), by programming of a digital computer, let it be noted that the pulse transfer function of the compensator can be realized in at least two other ways if a digital computer is not available or economically feasible. /2/ First is the realization by a delay-line network, sometimes referred to as a pulsed-data processing unit, through the use of delay elements, potentiometers, and summing amplifiers. The pulse transfer function can also be realized by a pulsed-data network consisting of resistors and capacitors. Inductors are avoided, because in the usual frequency range of digital and sampled-data control systems the sizes and weights of the required inductors would be prohibitive and the losses would be excessive. Pulsed data RC networks have the advantages of simplicity in structure and economy in implementation.

When a digital computer is available, realization is relatively simple by programming. The program can generally be carried out by means of one of three different methods:

a. Direct programming

b. Iterative, or cascade, programming

c. Parallel programming

Let the pulse transfer function (or Z-transform) of the digital controller be

$$D(z) = \frac{EOUT(z)}{EIN(z)} = \frac{\sum_{K=0}^{N-1} A_K z^{-K}}{1 + \sum_{j=1}^{M-1} B_j z^{-j}}$$

The first term in the denominator of D(z) must be a non-zero constant for the sake of physical realizability. In the direct method of programming it is necessary that this term be unity, since the direct method employs the difference equation equivalent of the above pulse transfer function. Writing the difference equation we get

$$EOUT[nT] = \sum_{K=0}^{N-1} A_K\, EIN[(n-K)T] - \sum_{j=1}^{M-1} B_j\, EOUT[(n-j)T]$$

In the digital computer this method requires 2M+2N-1 cells of storage for data and constants, M+N-1 multiplications, M+N-2 additions and/or subtractions, and M+N-2 data transfers.

The cascade method is based on the fact that D(z) is factorable into its poles and zeros such that a cascade of first order transfer function can be formed. In the computer this method requires 2M+N+3 cells of total storage, M+N+1 multiplications, M+N additions and/or subtractions, and M+2 data transfers.

The parallel method of programming is based on the partial fraction expansion of D(z) and requires 3M+2 cells of storage, 2M multiplications, 2M-1 additions and/or subtractions, and M+2 data transfers./1/

Before a choice of methods can be made on the basis of storage requirements or computation time, D(z) must be known, since any advantage of one method over the others depends primarily on the number of terms in the numerator of D(z). Generally, other considerations prevail. For

example, in the cascade method each stored constant is either a pole or a zero of D(z), and thus this method lends itself readily to experimental manipulation of the pulse transfer function in a "cut and try" sense. If storage space is at a premium, as in a special purpose digital computer, the direct method enjoys the advantage that data transfers can be effected by a single command that processes all the data in a shift register.[1] In the simulation program described in Appendix 1 the direct method of programming the digital controller is used.

6. Implementation of a double-rate digital controller by direct programming methods.

Direct programming takes the difference equation equivalent of the pulse transfer function and interprets z^{-1} to be a delay of one computation period. Thus, in the single-rate controller both input and output discrete sequences are equally spaced in time, and the programming is straight-forward. The computer program for a double-rate controller will be described in terms of the various arithmetical and shifting operations that must take place and not in terms of any particular programming language. The method will be shown using an example, and can be easily extended to the case of an n-rate controller.

Suppose that the pulse transfer function of the double-rate controller is:

$$D(z_2) = \frac{A1 + A2 \cdot z_2^{-1} + A3 \cdot z_2^{-2} + A4 \cdot z_2^{-3} + A5 \cdot z_2^{-4} + A6 \cdot z_2^{-5}}{1 + B2 \cdot z_2^{-1} + B3 \cdot z_2^{-2} + B4 \cdot z_2^{-3} + B5 \cdot z_2^{-4} + B6 \cdot z_2^{-5}}$$

The output from the controller will be a sequence, $Eout(Z_2)$, of discrete values separated in time by T/2 seconds. The input sequence is spaced at intervals of T as determined by the error sampling rate. When we write the difference equation to find a new value for Eout by cross-multiplying the transfer function and treating Z_2^{-1} as a delay of T/2 seconds we get:

$$\begin{aligned} EOUT[N \cdot T/2] &= A1 \cdot EIN[N \cdot T/2] + A2 \cdot EIN[(N-1)T/2] + A3 \cdot EIN[(N-2)T/2] \\ &+ A4 \cdot EIN[(N-3)T/2] + A5 \cdot EIN[(N-4)T/2] + A6 \cdot EIN[(N-5)T/2] \end{aligned}$$

$$- \left\{ B2 \cdot E_{OUT}\left[(N-1)T/2\right] + B3 \cdot E_{OUT}\left[(N-2)T/2\right] + B4 \cdot E_{OUT}\left[(N-3)T/2\right] + B5 \cdot E_{OUT}\left[(N-4)T/2\right] + B6 \cdot E_{OUT}\left[(N-5)T/2\right] \right\}$$

Since the discrete sequence, Ein(z), is at intervals of T seconds only half of the terms involving Ein above will be non-zero. Which half depends on whether N is even or odd. When N is even (ie., an output value is generated at the "same" time the error is sampled) then the terms involving A1, A3, and A5 are non-zero. When N is odd (ie., an output value is produced by the controller between input samples) then the only terms with non-zero values of Ein are those involving A2, A4, and A6. This amounts to alternately multiplying the input sequence by the even numbered coefficients of the numerator of $D(Z_2)$, then the odd numbered ones, ...etc.

In a given period of time the output sequence from a double-rate digital controller will consist of twice as many discrete values as the input sequence. For the sake of illustration let us number both sequences backward in time in the same way, thus: The most recent value is number one, the next most recent value is number two, etc. The total number of values in each sequence that must be stored in the computer memory is determined by the number of terms in the controller transfer function. In the example of this section we require output values numbers one through six and input values one, two and three. Each output value is employed only once in each numbered position

and passes from storage after six intervals of T/2. Each input value is employed <u>twice</u> in each numbered position; once as a multiplier of A_j and T/2 seconds later as a multiplier of A_{j+1}. Since the numerator of $D(Z_2)$ contains six terms, each input value also passes from storage after six intervals of T/2. A simple-minded pictorial representation may serve to clarify these ideas. Let the present time be the 16th instant of the closing of the output sampling switch of the double-rate controller. Neglecting computation delay (which is usually at least an order of magnitude less than the sampling period) the present time is also the eighth closing of the input sampling switch. The input and output sequences may be thought of as existing in storage as shown in Figure 7.

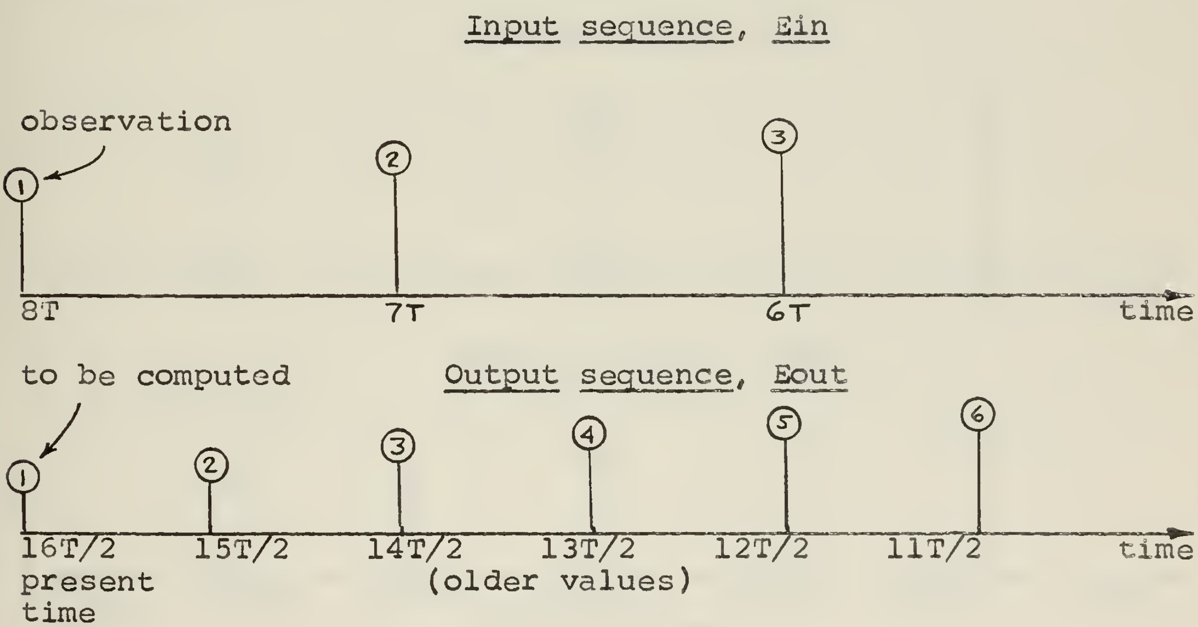


Figure 7. Input and output sequences for $D(Z_2)$

In the difference equation the terms involving the input

sequence would be (since in this example, N = 16):

A1•Ein(8T) or A1•Ein(1)

A3•Ein(7T) or A3•Ein(2)

A5•Ein(6T) or A5•Ein(3)

The value of Ein(1) has just been measured by the input sampler and the controller produces Eout(1).

Now let us move on in time by T/2 seconds. It is necessary to re-number the output sequence since a new Eout(1) will be produced. The previous Eout(1) becomes the current Eout(2), etc., with the previous Eout(6) being discarded. We are now <u>between</u> input samples and no new input information is to be received. The previous input values have become T/2 seconds older, however, and the sequences in storage might be pictured as shown in Figure 8.

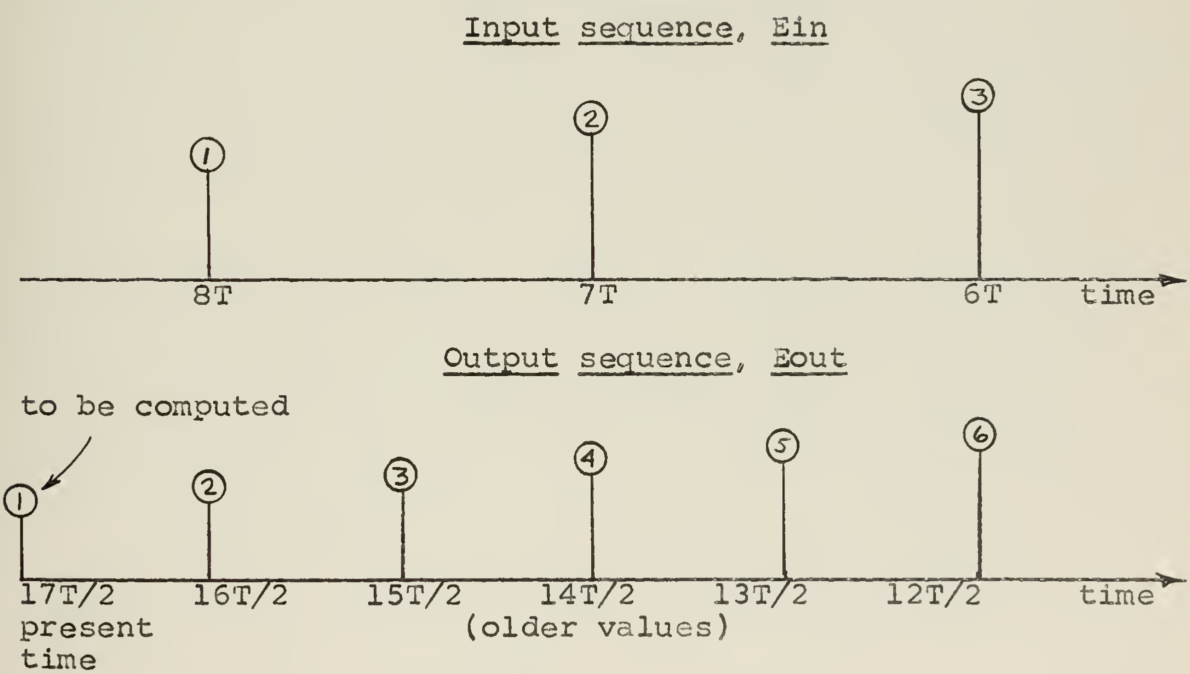


Figure 8. Sequences between input samples

Since the difference equation is in terms of T/2 the non-zero

terms involving the input sequence have changed and are now (for N = 17):

A2·Ein(8T) or A2·Ein(1)

A4·Ein(7T) or A4·Ein(2)

A6·Ein(6T) or A6·Ein(3)

The controller computes a new output value using all of the denominator coefficients of its transfer function but only the even-numbered numerator coefficients. As before, the output sequence must be shifted since we are only half a period away from receiving a new observed value for Ein(1).

For an example of the implementation of these ideas using the CDC Fortran programming language see Appendix 1.

7. Applications.

The results to be described in this section were obtained using the digital computer program simulation of a control system described in Appendix 1. The response curves were obtained through use of a Graphplot computer routine for the CDC 160 digital computer devised by Lt. Robert L. Hogg of the U. S. Naval Postgraduate School, Monterey, California. The program in Appendix 1 also produces numerical results on a printer, but these are not included in this work.

Initial investigation was made of a simple plant described by a second order differential equation whose pulse transfer function had no zero outside the unit circle in the z-plane. Next, a third order plant with a pulse transfer function containing a zero outside the unit circle was investigated. Both of these plants were theoretical and in neither case was an actual physical counter-part at hand.

At the U. S. Navy Electronics Laboratory, San Diego, California, a group headed by John B. Slaughter is engaged in studies of Digital Sampled-Data Feedback Systems for Shipboard Control Applications (NEL problem N4-17). The author was privileged to spend ten weeks with that group early in 1963, during which time a twin 3"/50 gun mount was in the process of being installed to serve as the plant in an experimental control system. A USQ-20 general purpose digital computer is to be employed as the controller. The

manufacturers of the gun mount supplied a fifth order transfer function for the gun in train. The majority of the simulation studies to be described in this thesis were conducted using this transfer function, and it is felt that the results obtained are applicable not only to this particular obsolescent plant but also to present day shipboard weapons, especially missile launchers.

The simulation program has the capability of testing a system for input signals of either unit step, ramp, or acceleration. In addition, additive Gaussian noise of selected mean and variance may be included in the input. The first action of the program is to produce the response obtained in the case of the continuous, un-compensated system for purposes of comparison. In the event this response is unstable no graph is made of it. Instead, the test input is reproduced. The next action of the program is to simulate the same plant incorporated in a sampled-data system with a single-rate digital controller and produce the response for this case. The program is further capable of simulating a system using a double-rate digital controller to produce a third response curve, although this is not done in every experiment. For convenience, the program will also compute the pulse transfer function of the controller when requested to do so, or the controller coefficients may be provided by the user. The Z-transforms employed in this work were computed by the program described in Appendix 2 and were verified by hand for the second and third order systems.

The second order plant has a Laplace transfer function given by
$$H(s) = \frac{1147}{s(s+25)}$$

The pulse-transfer function of the plant and zero-order hold combination for a sampling period of T = .01 seconds is

$$G(z) = \frac{.0528 z^{-1}(1+0.920455 z^{-1})}{(1-z^{-1})(1-0.778801 z^{-1})} = \frac{.0528 z^{-1} + 0.0486 z^{-1}}{1-1.778801 z^{-1} + 0.778801 z^{-1}}$$

There are no zeros of G(z) on or outside the unit circle in the z-plane, so minimal prototype compensation can be achieved. As noted earlier, the minimal prototype over-all pulse transfer function for a step input is $K(z) = z^{-1}$, so

$$D(z) = \frac{1}{G(z)} \cdot \frac{K(z)}{1-K(z)} = \frac{1-1.778801 z^{-1} + .778801 z^{-2}}{.0528 z^{-1} + .0486 z^{-1}} \cdot \frac{z^{-1}}{1-z^{-1}}$$

$$= \frac{\cancel{(1-z^{-1})}(1-.778801 z^{-1})}{.0528\cancel{z^{-1}}(1+.920455 z^{-1})} \cdot \frac{\cancel{z^{-1}}}{\cancel{1-z^{-1}}} = \frac{18.939394 - 14.750019 z^{-1}}{1.0 + 0.920455 z^{-1}}$$

(note that 1-K(z) contains the pole of G(z) lying <u>on</u> the unit circle.)

Figure 9 shows the plant behaviour in a continuous, uncompensated system and the behaviour in a sampled-data system using the digital controller above. Note that there is zero systematic error at the first sampling instant after the input was applied, and also at all succeeding sample instants. For a K(z) containing "n" terms there will be zero error for a design input after "n" sampling instants. The plant behaviour <u>between</u> samples is another matter, and in this example probably would not be considered satisfactory, unless the only desire was to increase the speed of

response with no regard for over-shoot and inter-sample ripple.

Simulated Gaussian noise with a standard deviation of 0.204 was added to the input and the experiment was repeated. Figure 10(a) shows the results. According to Monroe the process should have unity variance reduction factor since the sum of the squares of the process coefficients is one. Judging from Figure 10(b), which shows the noisy test input, this is approximately correct, since no noise reduction is apparent.

In an effort to improve the transient behaviour for a step input a staleness factor of 0.5 was selected. The over-all process was designed from

$$K(z) = \frac{a \cdot K(z)_{minimal}}{1 - 0.5z^{-1}}$$

and the Final Value Theorem showed that for zero steady-state error, $a = 1 - c = 0.5$ so that $K(z) = \dfrac{0.5z^{-1}}{1 - 0.5z^{-1}}$

$$\therefore D(z) = \frac{(1-z^{-1})(1-0.778801z^{-1})}{.0528z^{-1} + .0486z^{-1}} \cdot \frac{\dfrac{0.5z^{-1}}{1-0.5z^{-1}}}{1 - \dfrac{0.5z^{-1}}{1-0.5z^{-1}}}$$

$$= \frac{0.5 - 0.3894005z^{-1}}{.0528 + .0486z^{-1}} = \frac{9.469697 - 7.3750095z^{-1}}{1.0 + 0.920455z^{-1}}$$

The step response using this digital controller is shown in Figure 11. This is an example of a non-finite settling time process, but for practical purposes zero error is achieved at about the seventh sampling instant and thereafter, and the inter-sample ripple and over-shoot is much

reduced compared to the minimal design.

This same staleness factor design was subjected to a noisy step input with the results shown in Figure 12. Since K(z) can be expressed as an infinite series by long division, thus: $0.5z^{-1} + .25z^{-2} + .125z^{-3} + .0625z^{-4}....$ the variance reduction factor is approximately one-third. It is not apparent from the simulation results that the process indeed had this noise reducing property.

The six point finite memory process for unity variance reduction (see page 23 above) was then subjected to a step input with and without noise. Figures 13 and 14 show the results. This process and the following one depend on the pulse transfer function of the plant not having a zero outside the unit circle. It can be seen in Figure 13 that the transient behaviour of a design whose sole criterion is variance reduction is rather poor. For this reason the next process considered was a composite one in which three-quarters of the design effort was towards transient control and only one-quarter towards noise reduction. The program on page 27 was used to compute the process coefficients, as shown on the same page. This gave the most complex controller so far with 11 numerator terms and 11 denominator terms. The results of the simulation experiment with this controller are shown in Figures 15 and 16. The transient behaviour displayed in Figure 15 is a noticeable improvement over that of Figure 13 where the only aim was noise reduction.

At this point a certain disenchantment with the notion

of noise reduction begins to set in. The plant is a simple, well-behaved one, and in the four experiments so far using "clean" input signals both the continuous system responses and the sampled-data system responses can be verified analytically with some effort. When the unit step input has "white" noise added to it as shown in Figure 10(b) two tentative conclusions suggest themselves. First, the continuous system with no compensation is relatively unaffected by the presence of the noise. A close inspection of Figure 10(a) reveals that the peak overshoot still occurs at about one-tenth of a second and its value is increased only slightly (from 29% to 37%). A steady state error equal to the mean squared value of the noise can be inferred from Figures 12 and 16 in which the time scale has been doubled for a longer look at the plant's response. Over-all, however, the low-pass character of the plant effectively prevents any drastic reaction to the presence of the noise. Not so in the case of the sampled-data system, so our second thought at this point is that the insertion of the digital controller in the forward path disrupts the low-pass filtering action of the plant. The noise in the error signal is the same as in the input, and the noise power in each discrete error sample is integrated over the next sampling period and applied to the plant. This integrated noise power causes some fairly drastic gyrations in the plant regardless of the design criteria. A brief comparison of the four noisy experiments up to this point provides small reason for employing a variance reduction design in

preference to a noise-free criterion. It appears that the effort could better be expended in pre-filtering or other measures to eliminate noise prior to the input to a sampled-data system.

The next four simulation studies were conducted using a third order plant whose transfer function is given by

$$H(s) = \frac{80{,}000}{s(s+25)(s+70)}$$

For T = .01 seconds the z-transform of the plant-and-hold is

$$G(z) = \frac{.010600z^{-1} + .033672z^{-2} + .006584z^{-3}}{1.0 - 2.275386z^{-1} + 1.662127z^{-2} - 0.386741z^{-3}}$$

or in factored form

$$G(z) = \frac{.0106z^{-1}(1+0.209327z^{-1})(1+2.967275z^{-1})}{(1-z^{-1})(1-0.778801z^{-1})(1-0.496585z^{-1})}$$

The presence of a zero of G(z) outside the unit circle is noted. This zero will prevent us from realizing a "minimal" prototype response. In accordance with the rules given on page 16 the design of a process giving prototype response to a step input proceeds as follows:

$$K(z) = (1+2.967275z^{-1})a_1 z^{-1}$$

$$1-K(z) = (1-z^{-1})(1+b_1 z^{-1})$$

Solving for the two unknowns, a_1 and b_1, results in

$a_1 = .252062 \qquad b_1 = .747938$ and thus

$$K(z) = 0.252062z^{-1} + 0.747938z^{-2}$$
$$= (1+2.967275z^{-1})0.252062z^{-1}$$

Solving for the digital filter coefficients by hand, we write

$$D(z) = \frac{\cancel{(1-z^{-1})}(1-.778801z^{-1})(1-.496585z^{-1})\cancel{(1+2.967275z^{-1})}.252062\cancel{z^{-1}}}{(.0106)(1+.209327\cancel{z^{-1}})\cancel{(1+2.967275z^{-1})}\cancel{(1-z^{-1})}(1+.747938z^{-1})}$$

and after cancellations of the factors shown, we get

$$D(z) = \frac{23.779434 - 30.327983z^{-1} + 9.196591z^{-2}}{1.0 + 0.957267z^{-1} + 0.156562z^{-2}}$$

The un-compensated response and the response of the sampled-data system using the D(z) above are shown in Figure 17. The non-minimal nature of the digitally-controlled response is characterized by the fact that it took two sampling periods (.02 seconds) for the output to reach zero error at sampling instants. There is a slight amount of inter-sample ripple, but the peak overshoot is greatly reduced from what it was for the continuous system. The improved damping of this third order sampled-data system compared to the second order system considered previously is attributed to additional filtering action of the pole at -70. Since this design was for a step input there will not be zero systematic error for higher order inputs, such as a ramp. This can be observed in Figure 18 where the curve labelled D1 is the response of the digitally controlled system to a ramp input. (Note: in this figure, and all others for which the test input is a ramp, a 45 degree slope from the origin is the input signal). The controlled response settles down to a steady state error of about .017 units after two sample periods. Whether or not this is an intolerable error in response to a velocity input depends on the system. For shipboard fire-control application such as tracking a target, a

design for zero error to a ramp is usually preferable. More about this later.

An additional refinement of the step response of this third order system can be achieved at the cost of one extra period, T, in reaching steady state by designing for ripple-free response as discussed on page 17. For this case we get

$$K(z) = (1 + 0.209327 z^{-1})(1 + 2.967275 z^{-1}) a_1 z^{-1}$$

$$1 - K(z) = (1 - z^{-1})(1 + b_1 z^{-1} + b_2 z^{-2})$$

and by equating coefficients of like powers of z^{-k} $(k=1,2,3)$ we obtain the below equations to solve for a1, b1, and b2.

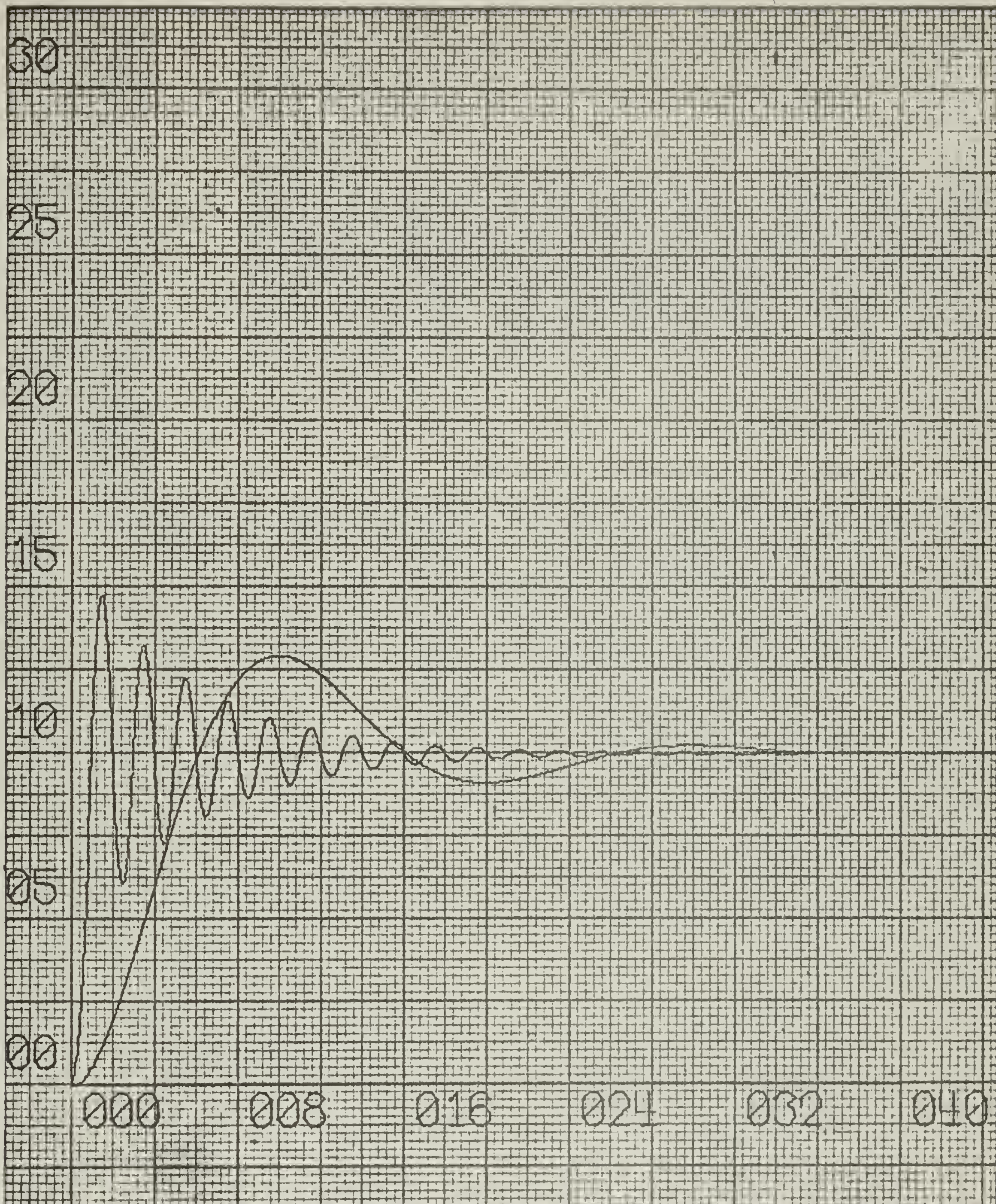
$$a_1 = 1 - b_1$$

$$-3.176602 a_1 = b_2 - b_1$$

$$0.6211308 a_1 = b_2$$

Solving these, we get al = .2084318 so that the process is $K(z) = .2084318z^{-1} + .662105z^{-2} + .1294634z^{-3}$

These process coefficients as well as the pulse transfer function of the plant were provided to the simulator program which then calculated the digital controller and produced the ripple-free response shown in Figure 19. Figure 20 shows the same system with a velocity input. The steady state ramp error has increased slightly, from .017 to .020 due to the additional term in the process z-transform, and the controller had a total of six terms. In the absence of input noise we have demonstrated satisfactory digital control of this lightly damped plant when the input is a unit step.



30
25
20
15
10
05
00
000
008
016
024
032
040
X AXIS SCALE = 8.00E-02
Y AXIS SCALE = 5.00E-01
SECOND-ORDER PLANT
MINIMAL PROTOTYPE DESIGN FOR STEP
NOISE-FREE STEP INPUT

Figure 9

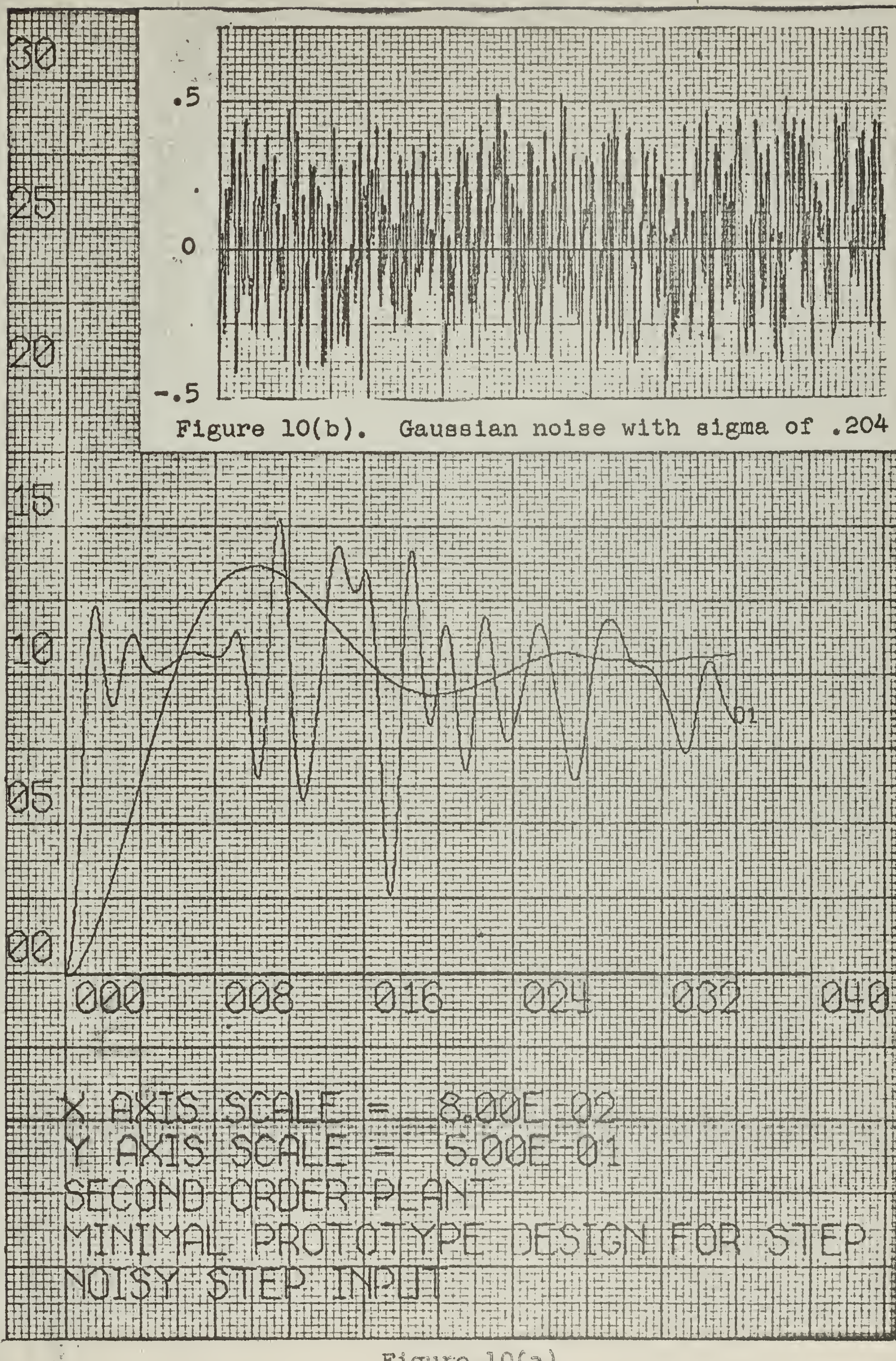
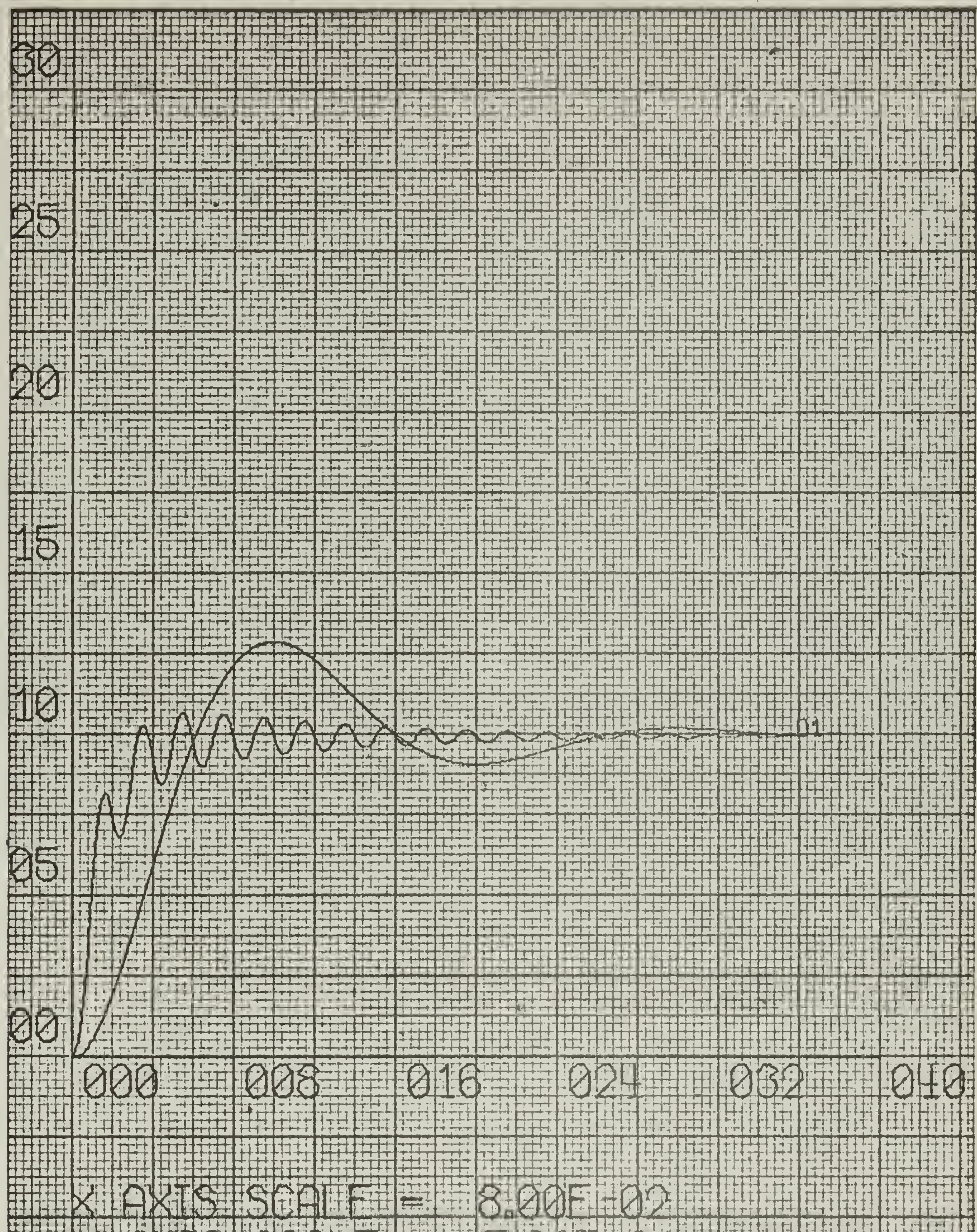


Figure 10(b). Gaussian noise with sigma of .204

Figure 10(a)



30
25
20
15
10
05
00
000
008
016
024
032
040
X AXIS SCALE = 8.00E-02
Y AXIS SCALE = 5.00E-01
SECOND ORDER PLANT
STALENESS FACTOR OF 0.5
NOISE-FREE STEP INPUT

Figure 11

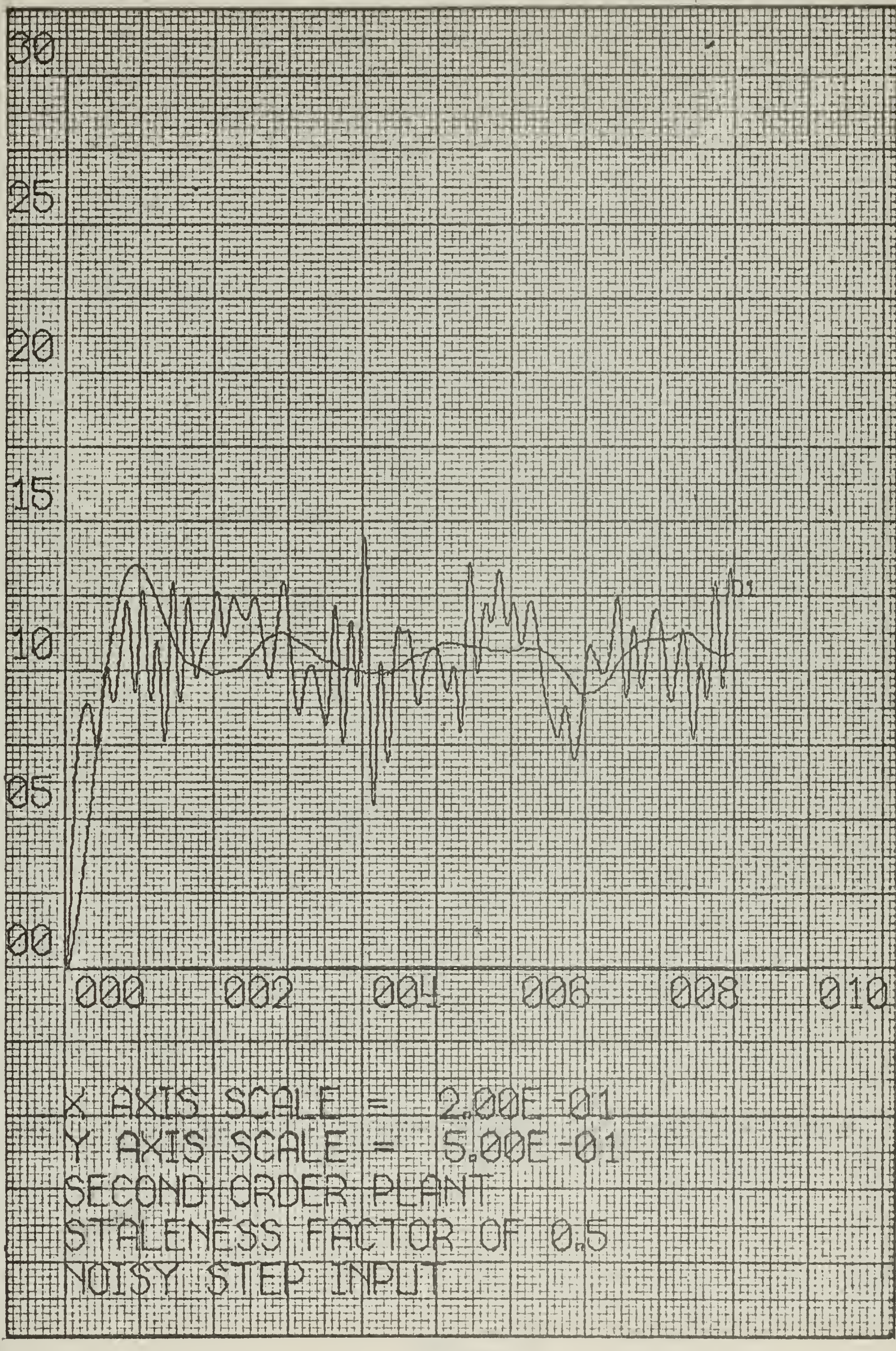

30
25
20
15
10
05
00
000
002
004
006
008
010
X AXIS SCALE = 2.00E-01
Y AXIS SCALE = 5.00E-01
SECOND ORDER PLANT
STALENESS FACTOR OF 0.5
NOISY STEP INPUT

Figure 12

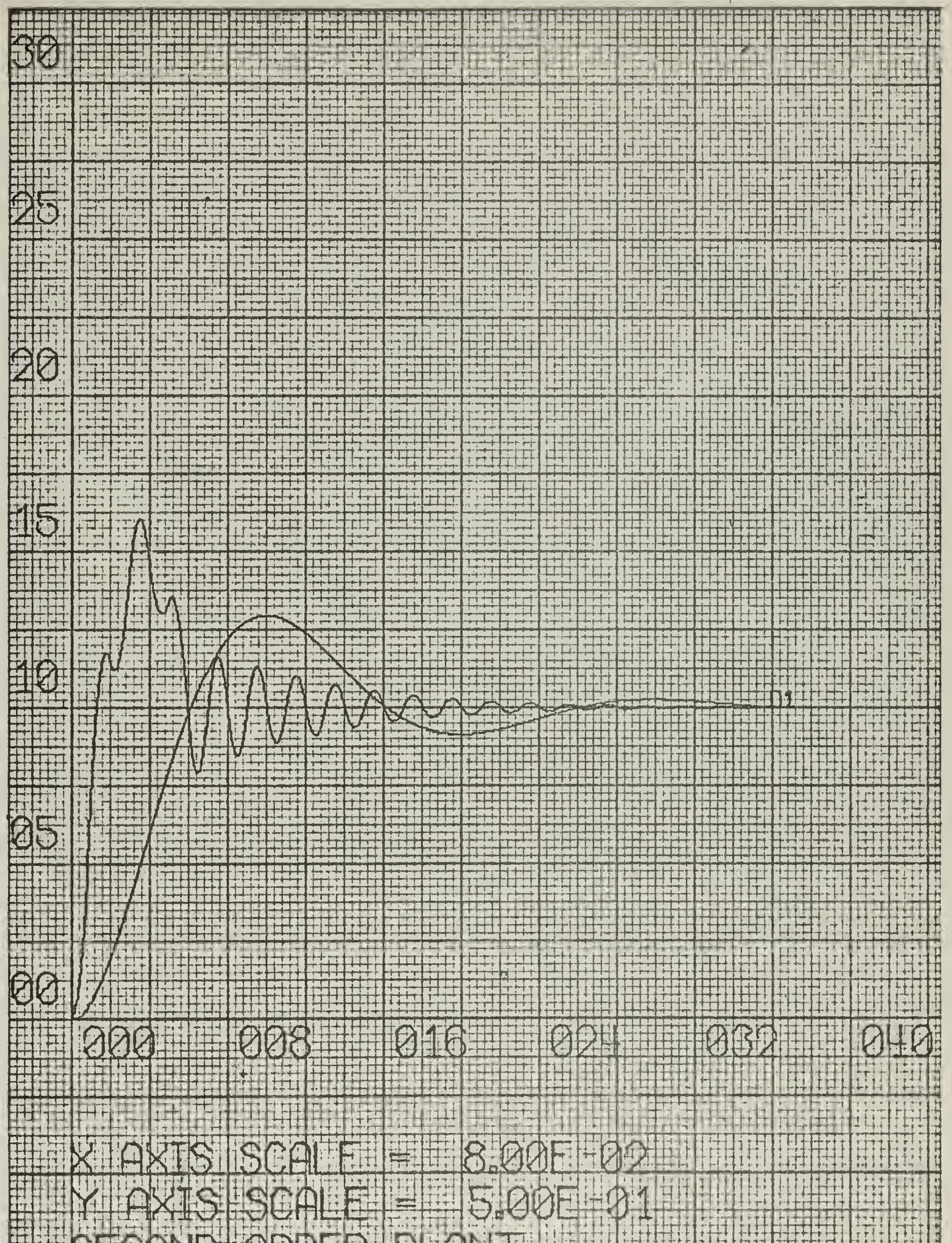

30
25
20
15
10
05
00
000
008
016
024
032
040
01
X AXIS SCALE = 8.00E-02
Y AXIS SCALE = 5.00E-01
SECOND ORDER PLANT
UNITY VARIANCE DESIGN
NOISE-FREE STEP INPUT

Figure 13

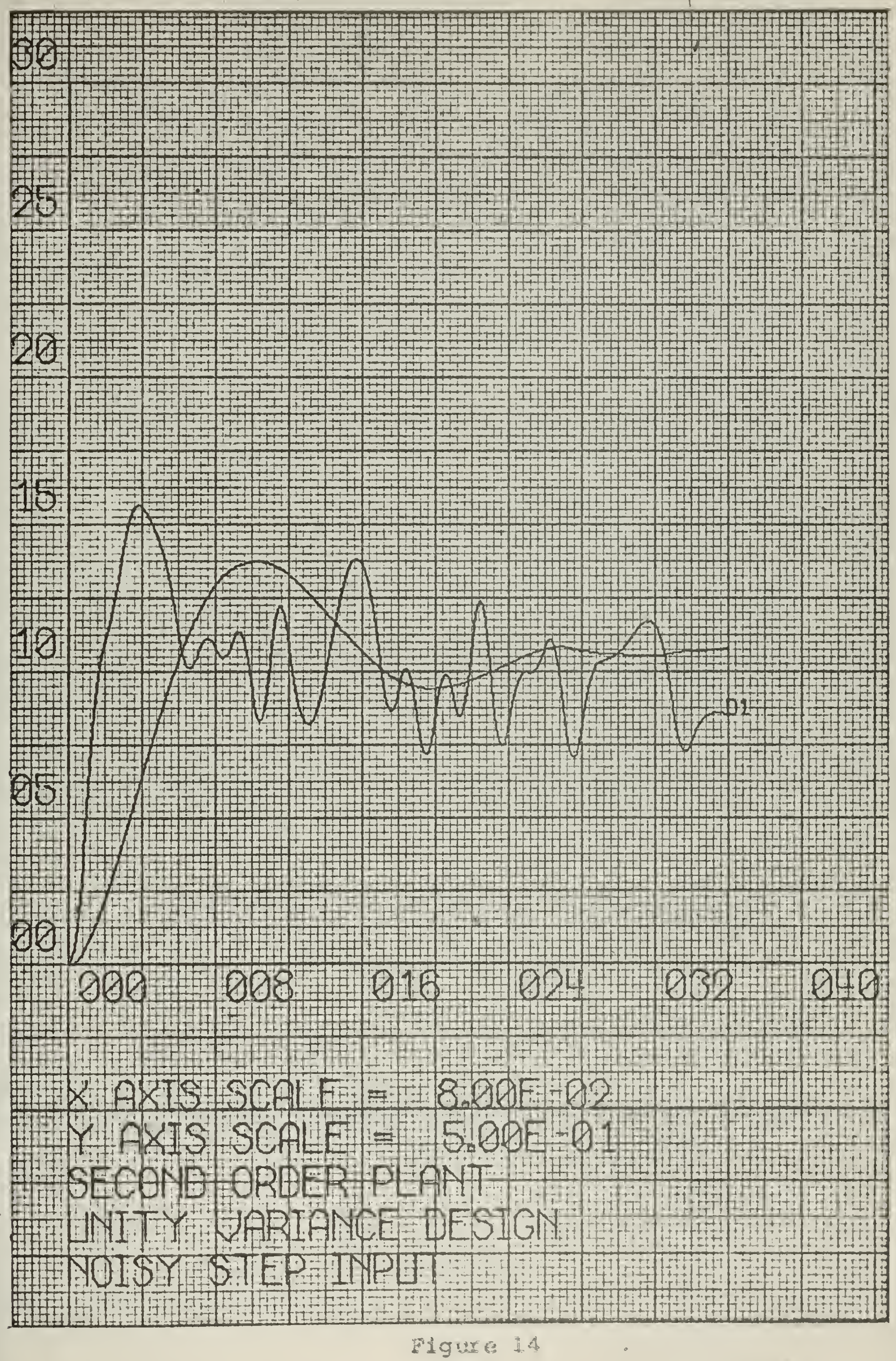

30
25
20
15
10
05
00
000
008
016
024
032
040
D1
X AXIS SCALE = 8.00E-02
Y AXIS SCALE = 5.00E-01
SECOND ORDER PLANT
UNITY VARIANCE DESIGN
NOISY STEP INPUT

Figure 14

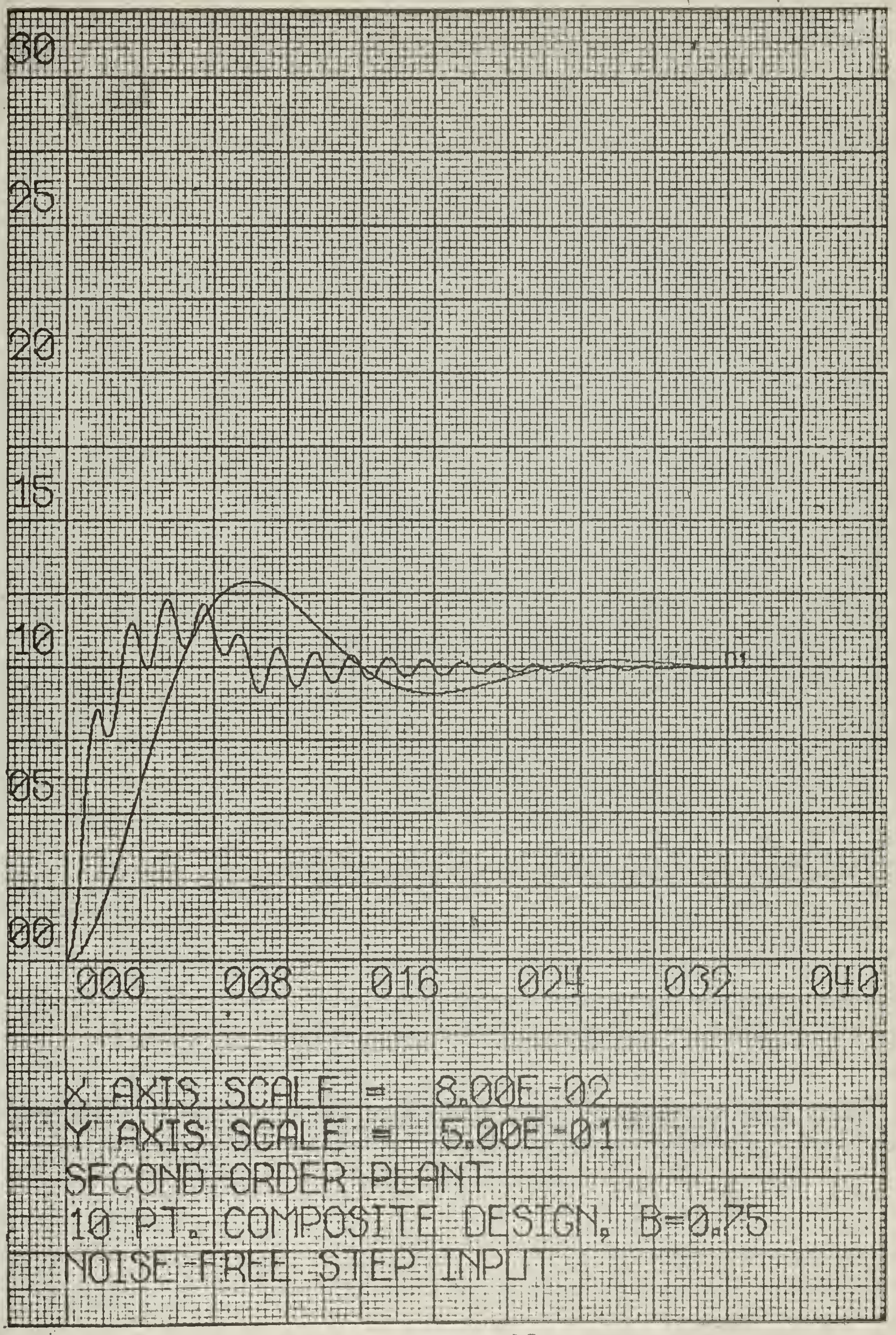

30
25
20
15
10
05
00
000
008
016
024
032
040
X AXIS SCALE = 8.00E-02
Y AXIS SCALE = 5.00E-01
SECOND ORDER PLANT
10 PT. COMPOSITE DESIGN, B=0.75
NOISE FREE STEP INPUT

Figure 15

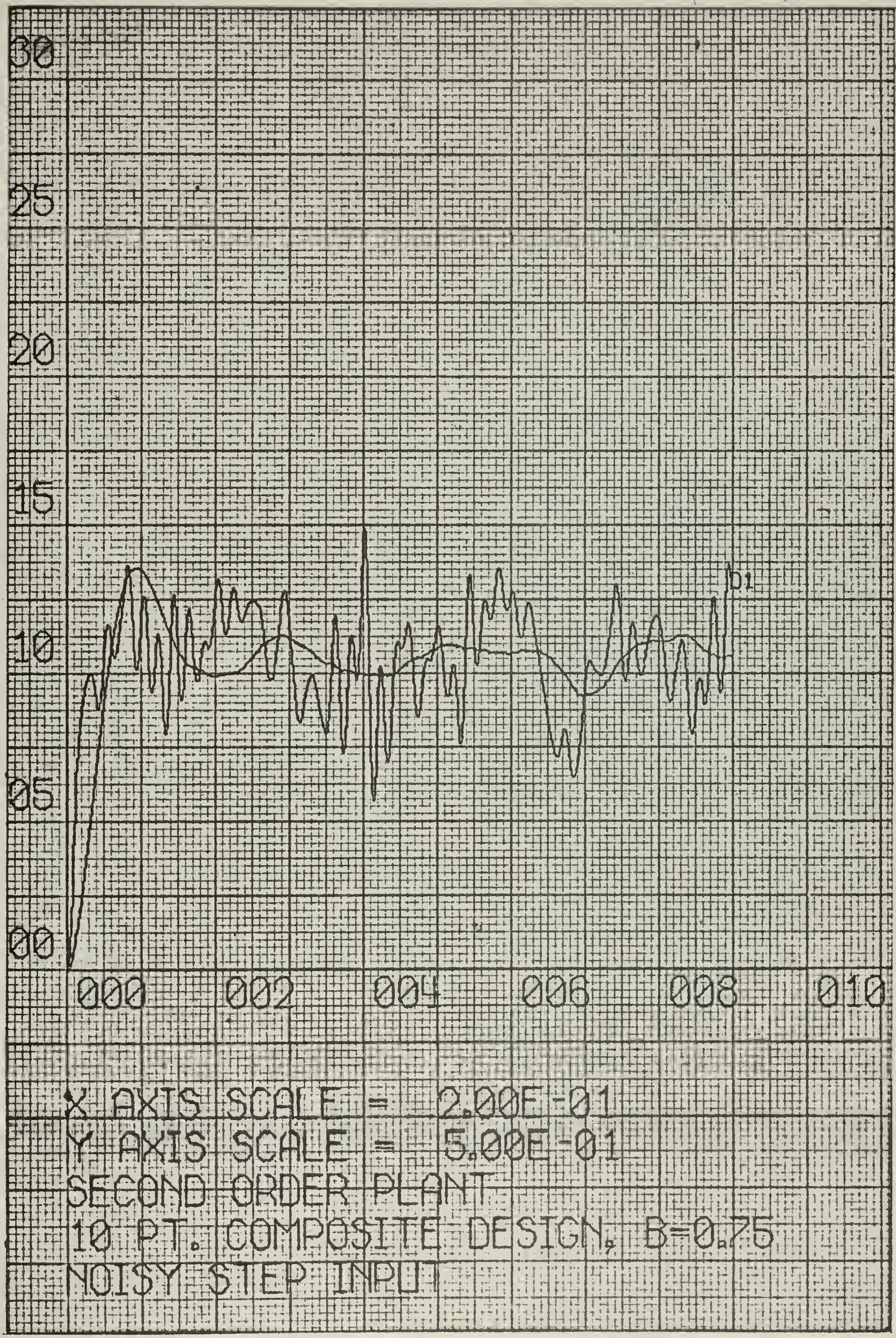

30
25
20
15
10
05
00
000
002
004
006
008
010
b1
X AXIS SCALE = 2.00E-01
Y AXIS SCALE = 5.00E-01
SECOND ORDER PLANT
10 PT. COMPOSITE DESIGN, B=0.75
NOISY STEP INPUT

Figure 16

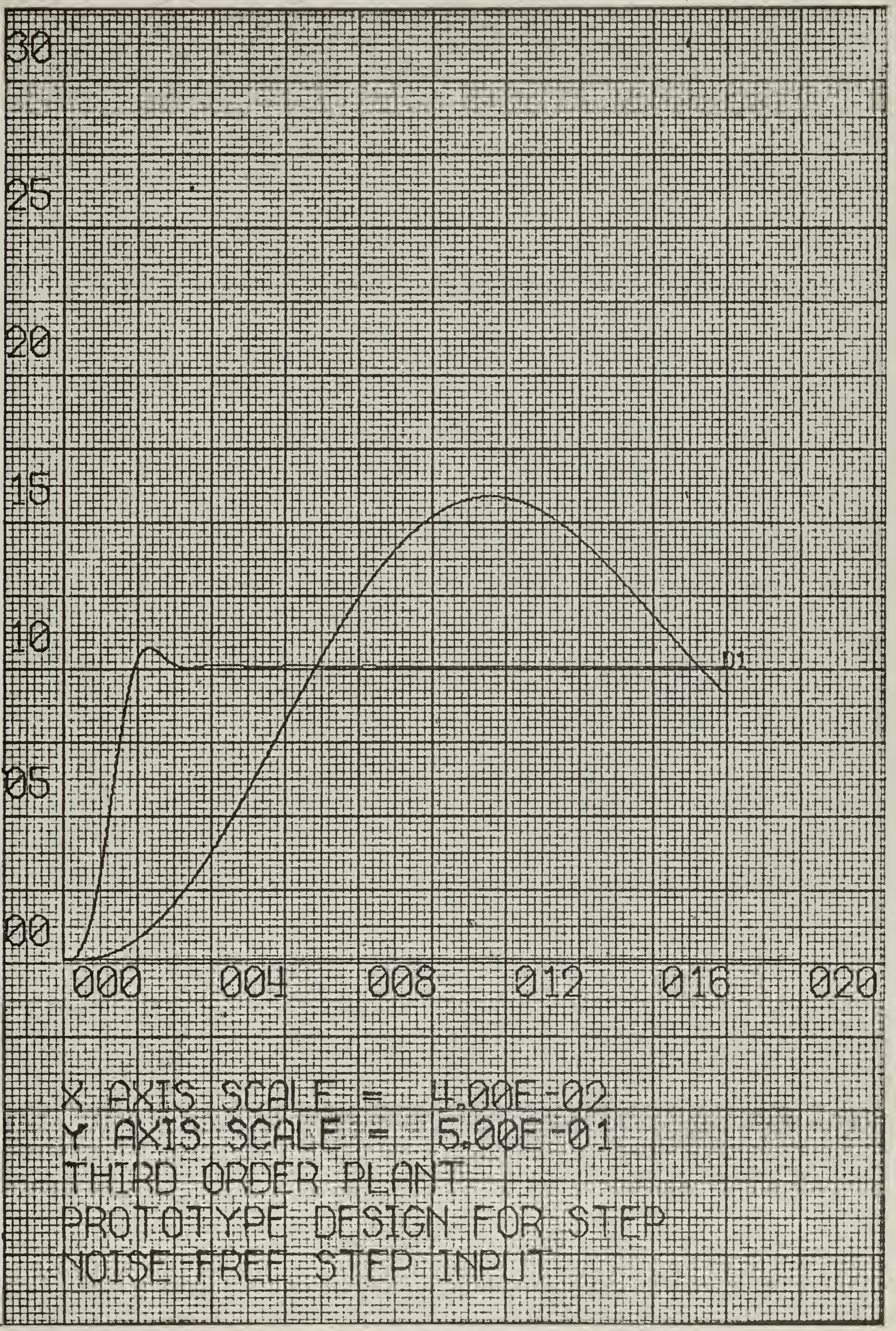

30
25
20
15
10
05
00
000
004
008
012
016
020
D1
X AXIS SCALE = 4.00E-02
Y AXIS SCALE = 5.00E-01
THIRD ORDER PLANT
PROTOTYPE DESIGN FOR STEP
NOISE FREE STEP INPUT

Figure 17

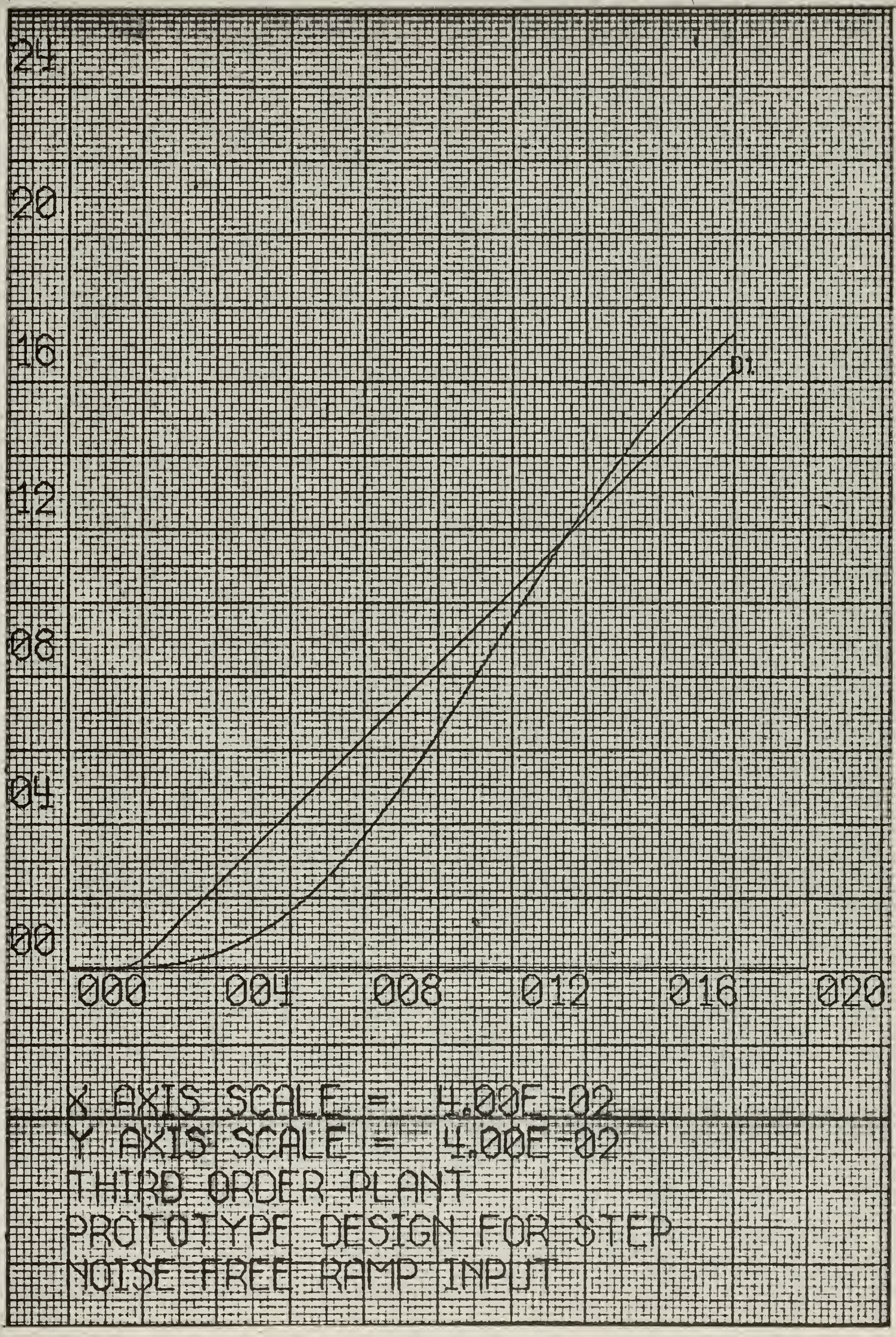

24
20
16
12
08
04
00
01
000
004
008
012
016
020
X AXIS SCALE = 4.00E-02
Y AXIS SCALE = 4.00E-02
THIRD ORDER PLANT
PROTOTYPE DESIGN FOR STEP
NOISE-FREE RAMP INPUT

Figure 18

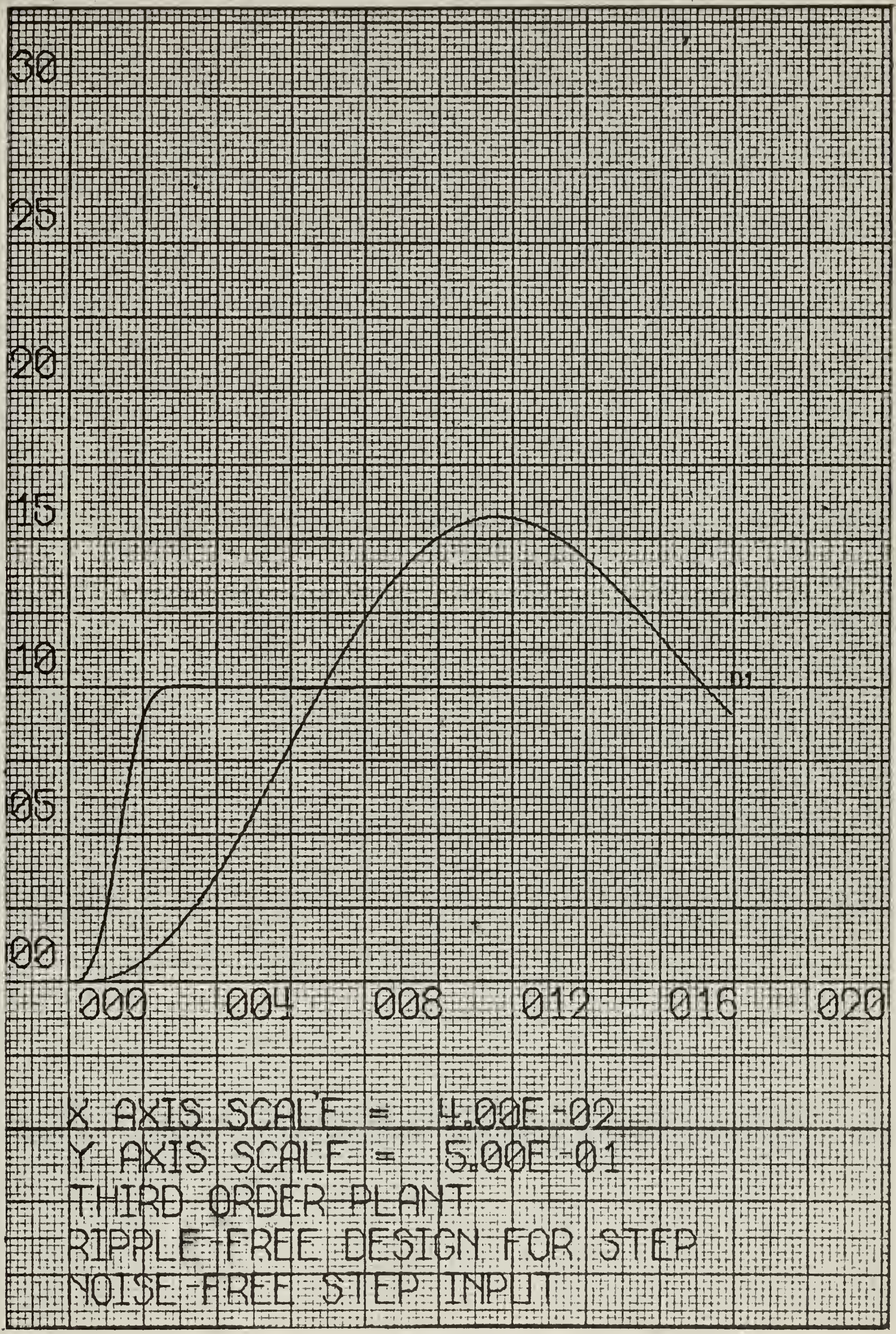

30
25
20
15
10
05
00
000
004
008
012
016
020
D1
X AXIS SCALE = 4.00E-02
Y AXIS SCALE = 5.00E-01
THIRD ORDER PLANT
RIPPLE-FREE DESIGN FOR STEP
NOISE-FREE STEP INPUT

Figure 19

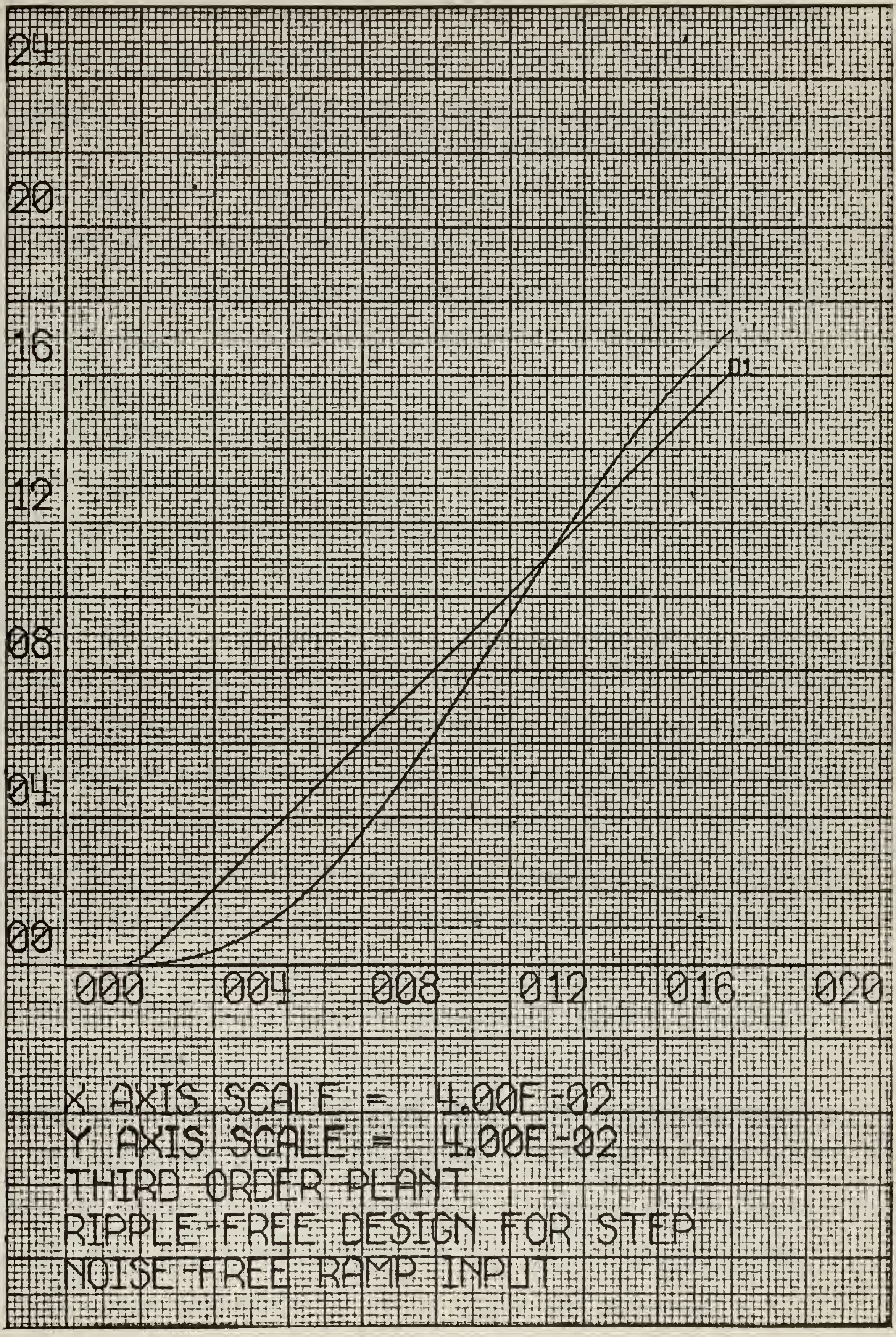

24
20
16
12
08
04
00
000
004
008
012
016
020
01
X AXIS SCALE = 4.00E-02
Y AXIS SCALE = 4.00E-02
THIRD ORDER PLANT
RIPPLE-FREE DESIGN FOR STEP
NOISE-FREE RAMP INPUT

Figure 20

Turning now to the twin 3"/50 gun mount, the transfer function of the forward path train drive is given by

$$H(s) = \frac{1.6064235 \times 10^9}{s(s+11)(s+17)(s+23)(s+83)}$$

Most of this very large forward gain is supplied by the amplidyne drive units and results in a quickly responsive system when suitable compensation is added. In actual use the train drive function was elaborately compensated by various continuous networks in a multitude of feedback paths in order to obtain stable operation. All of this continuous compensation has been removed from the gun by the group at NEL, and the necessary control will be achieved by digital methods. Figures 21 and 22 were provided by Donald Lackowski, an engineer in that NEL group, and show the root locus and Routh criterion analysis for the forward transfer function with unity feedback. It can be seen that the stability limit on the gain, K, is roughly three orders of magnitude lower than the actual value present, and thus the plant is highly unstable without compensation. This is borne out by the simulation studies to be described next. In every case the continuous response went rapidly off scale and could not be shown in the figures to follow. As previously mentioned, in this event the simulator program reproduced the test input in lieu of the unstable response.

The computer program described in Appendix 2 was exceedingly helpful at this point since finding the z-transform of this fifth order equation for just a single value of T is quite laborious by hand. With the computer a

number of transforms was quickly obtained for a spectrum of sampling rates ranging from one hundred samples per second to as slow as one sample every four seconds. For the faster rates the pulse transfer functions themselves were of the fifth order (ie., five numerator terms and six denominator terms). As the rate was decreased it was observed that the higher ordered terms in both numerator and denominator of G(z) became progressively smaller. For T of one-half second or larger a good approximation of G(z) can be made using only a second order function. In addition, the zero of G(z) which lay outside the unit circle for smaller values of T had moved inside the circle for T = 0.5 seconds. The computer calculated pulse transfer function of the plant and hold combination for T of one-half second is given below:

Numerator of G(z)	Denominator of G(z)
	1.00000000
$1336.33153319z^{-1}$	$-1.00430037z^{-1}$
$897.96714053z^{-2}$	$.00430125z^{-2}$
$6.02702748z^{-3}$	$-.00000087z^{-3}$
$.00049220z^{-4}$	$.00000000z^{-4}$
$.00000000z^{-5}$	$.00000000z^{-5}$

The computer program also factored the numerator and gave the following real values for the zeros of G(z):

-.00000001, -.00008267, -.00669675, -.66518491, 0.0

It was decided to investigate the advantages of a multi-rate sampled data system, the analysis and design of which

was discussed beginning on page 17. The simulator program has the ability not only to produce results for the continuous system and a single-rate digitally controlled system, but also, upon request, to simulate a system using a digital controller whose output to the plant occurs twice as frequently as the input error samples are supplied to the controller. This is called a <u>double-rate</u> controller. The output rate was chosen to be two samples per second in order to take advantage of the second order approximation to G(z) mentioned above. It was desired that the over-all process have zero steady state error to a velocity (ramp) input. Following the method of page 19 the design for prototype ramp response proceeds as follows: The fundamental design equation for the double-rate controller is

$$D(z_2) = \frac{1}{G(z_2)} \cdot \frac{K(z_2)}{1 - K(z_2^2)}$$

For $T/2 = 0.5$ seconds, $G(Z_2) \approx \dfrac{1336\, z_2^{-1} + 898\, z_2^{-2}}{1 - z_2^{-1}}$

The Laplace transform of the input is $R(s) = \frac{1}{s^2}$, whence

$$R(z_2) = \frac{T/2 \cdot z_2^{-1}}{(1 - z_2^{-1})^2} \quad ; \quad R(z_2^2) = \frac{T \cdot z_2^{-2}}{(1 - z_2^{-2})^2} = R(z_1)$$

Applying steady state and realizability constraints, we get

$$K(z_2) = \frac{(1 - z_2^{-2})^2}{(1 - z_2^{-1})^2} \cdot F(z_2) \quad \text{where} \quad F(z_2) = a_1 z_2^{-1} + a_2 z_2^{-2} + \cdots$$

and a look ahead at the constraint equation,

$$\left. \frac{d^m Q(z_2)}{d(z_2^{-1})^m} \right|_{z_2 = 1} = 0 \qquad \text{for } m = 0, 1, \ldots, K-1$$

shows the need for only two free constants, a_1 and a_2, so

$$Q(z_2) = P(z_2) - P(z_2^2)\cdot F(z_2) = T/2 \cdot z_2^{-1} - T\cdot z_2^{-2}\left(a_1 z_2^{-1} + a_2 z_2^{-2}\right)$$

$$= 0.5 z_2^{-1} - z_2^{-2}\left(a_1 z_2^{-1} + a_2 z_2^{-2}\right) = 0.5 z_2^{-1} - a_1 z_2^{-2} - a_2 z_2^{-3}$$

$$\text{for } m = 0: \quad Q(z_2)\Big|_{z_2=1} = 0.5 - a_1 - a_2 = 0$$

$$\text{for } m = 1: \quad \frac{dQ(z_2)}{d z_2^{-1}}\Bigg|_{z_2=1} = 0.5 - 3a_1 - 4a_2 = 0$$

solving these equations we get: $a_1 = 1.5$, $a_2 = -1.0$

$$\text{so that } K(z_2) = \left(\frac{1 - z_2^{-2}}{1 - z_2^{-1}}\right)^2 \left(1.5 z_2^{-1} - z_2^{-2}\right) = \left(1 + z_2^{-1}\right)^2\left(1.5 z_2^{-1} - z_2^{-2}\right)$$

$$= 1.5 z_2^{-1} + 2.0 z_2^{-2} - 0.5 z_2^{-3} - 1.0 z_2^{-4}$$

To find $K(z_2^2)$ we take the even-powered terms with the result $K(z_2^2) = 2.0 z_2^{-2} - 1.0 z_2^{-4} = K(z_1)$ which can be recognized as the minimal prototype ramp response function shown on page 14 expressed in the Z_2 domain.

Substituting now in the fundamental design equation, the pulse transfer function of the double-rate controller is

$$D(z_2) = \frac{(1 - z_2^{-1})}{(1336 z_2^{-1} + 898 z_2^{-2})} \cdot \frac{(1.5 z_2^{-1} + 2.0 z_2^{-2} - 0.5 z_2^{-3} - 1.0 z_2^{-4})}{(1 - 2.0 z_2^{-2} + 1.0 z_2^{-4})}$$

$$= \frac{1.5 - z_2^{-1}}{1336 - 438 z_2^{-1} - 898 z_2^{-2}}$$

We must normalize this result with respect to the leading

denominator coefficient, since direct programming requires a value of unity in this position. The final result is

$$D(z_2) = \frac{.001122754 - .00074850\, z_2^{-1}}{1.0 - 0.32784431\, z_2^{-1} - 0.67215569\, z_2^{-2}}$$

Before proceeding with the simulation run we must provide for the single-rate action with T = 1.0 seconds. This is simply done by providing the simulator program with the ramp minimal prototype function and G(z) coefficients. For this sampling rate an even better second order approximation is available for G(z), namely:

$$G(z) = \frac{3576\, z^{-1} + 924\, z^{-2}}{1 - z^{-1}}$$

The responses of the two systems for a noise-free velocity input can be seen in Figure 23. The single-rate controller gave zero error at the second sampling instant and only moderate overshoot and inter-sample ripple. The double-rate system was 0.5 seconds faster but over-shoot was worse due to the larger control forces involved. The ripple in the double-rate case was more persistent, also.

These same systems were then subjected to step inputs with the results displayed in Figure 24. The highly tuned nature of any prototype design is exhibited here, where the input is of lower order than that for which the system was designed. An interesting result in Figure 24 which was not expected is the slightly decreased peak overshoot in the case of the double-rate controller.

The next design criterion applied was ripple-free

step response using both single and double-rate controllers and the same error-sampling rate of the preceding design. The ripple-free process, K(z), was computed and supplied to the simulator, which calculated the single-rate controller, D1. The double-rate controller was designed using the techniques discussed beginning at the bottom of page 20. Both controllers were quite simple and are given below:

$$D1 = \frac{.00022222}{1 + 0.2053333\, z^{-1}} \; ; \; D2 = \frac{.0004475 - .000001925\, z_2^{-1}}{1.0}$$

Figure 25 shows both systems responding without ripple to a noiseless step input. Again, the double-rate system is about one-half a second faster. The ramp response is next shown in Figure 26. Both systems have a steady state error since they were designed for a step input only. In the case of the double-rate controller the sustained ripple is due to the fact that the design technique was a special one, explicitly for ripple-free step response, and cannot be expected to give conventional behaviour for a different input.

The results of the simulation studies using the second order approximation designs with inputs contaminated with Gaussian noise tend to reinforce the tentative conclusions mentioned earlier. It was impossible to observe the response of the continuous system to a noisy input since the continuous system was unstable. The disturbing result is that the sample-data systems, both with single and double-rate

digital controllers, which had given stable performance of varying degrees of goodness when the input was noise-free were, with but a single exception, unstable in the presence of noise. The exception is shown in Figure 27 in which the design of D1 and D2 was for ripple-free response to a step and the input was a noisy step. It seems apparent at this point that a large signal-to-noise ratio is a *sine qua non* in the input to a sampled-data system.

In the preceding experiments it was only the pulse transfer function of the plant and hold combination that was approximated in order to simplify the design calculations. The differential equation of the plant itself remained of the fifth order for the numerical methods solution of all response values.

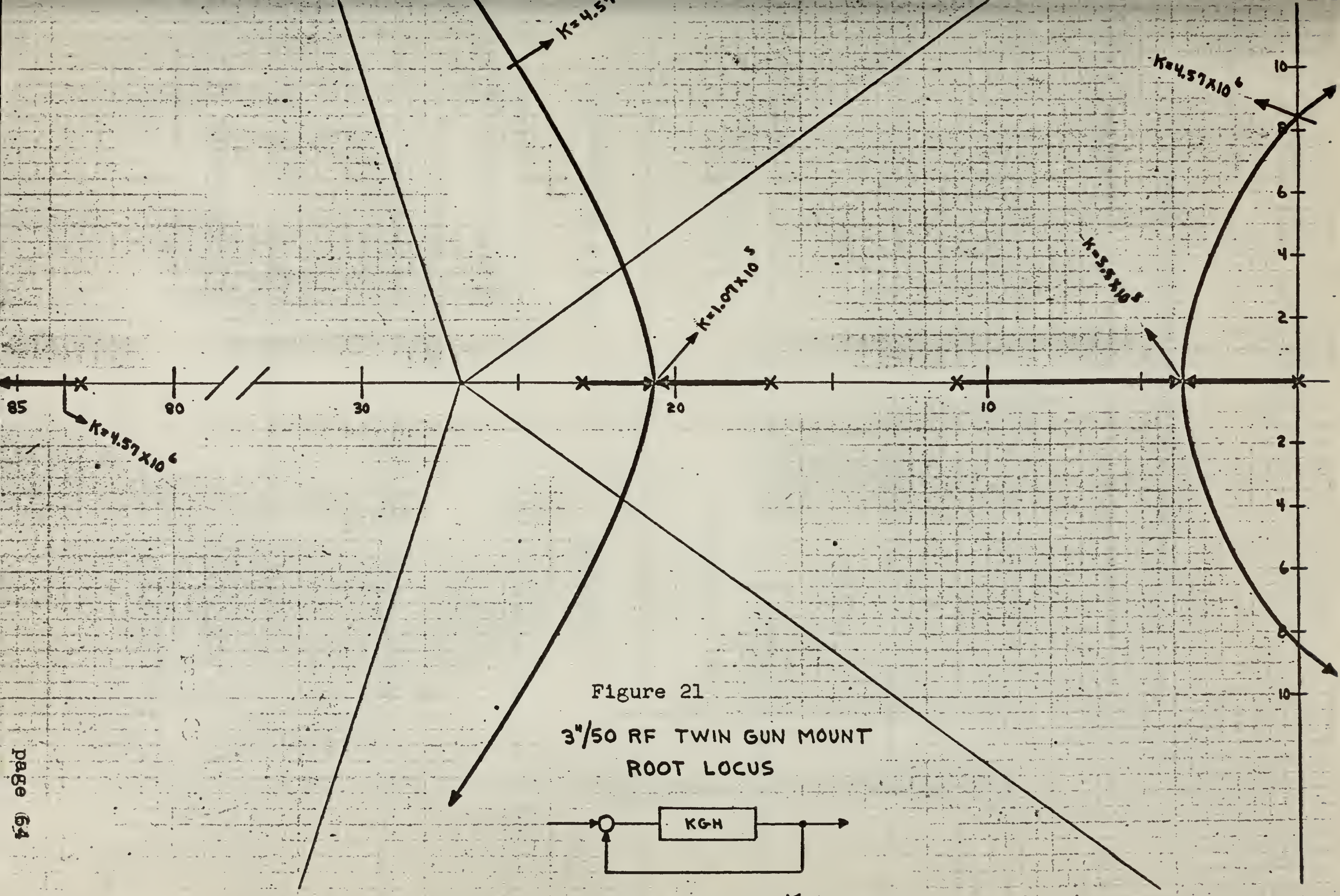


Figure 21

3"/50 RF TWIN GUN MOUNT
ROOT LOCUS

$$KGH = \frac{K}{s(s+11)(s+17)(s+23)(s+83)}$$

ROUTH'S STABILITY CRITERION:

HARACTERISTIC EQUATION:

$$s^5 + (134)s^4 + (5064)s^3 + (73274)s^2 + (356983)s + K = 0$$

OUTH'S ARRAY:

Figure 22

5	1	5064	356983
4	134	73274	K
3	605302	47835722 − K	0
2	$37942912 \times 10^3 + 134K$	605302K	0
	$-[67K^2 + 198961718228K - 907513295151232000]$	0	

FOR STABILITY: $0 < K < 4.574627 \times 10^6$

O DAMPING FREQ: $\omega = 8.47$

Figure 22

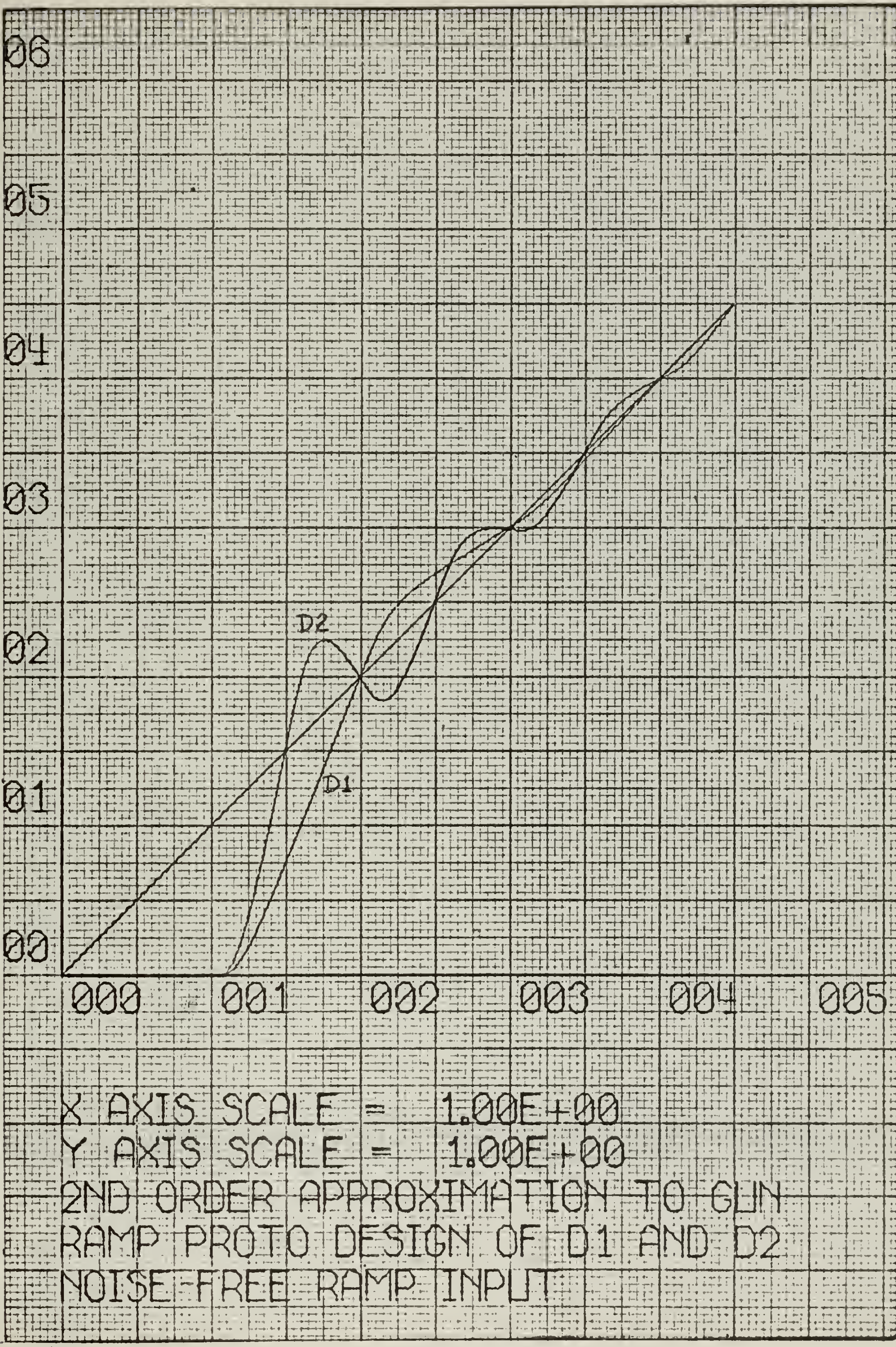

06
05
04
03
02
01
00
D2
D1
000
001
002
003
004
005
X AXIS SCALE = 1.00E+00
Y AXIS SCALE = 1.00E+00
2ND ORDER APPROXIMATION TO GUN
RAMP PROTO DESIGN OF D1 AND D2
NOISE FREE RAMP INPUT

Figure 23

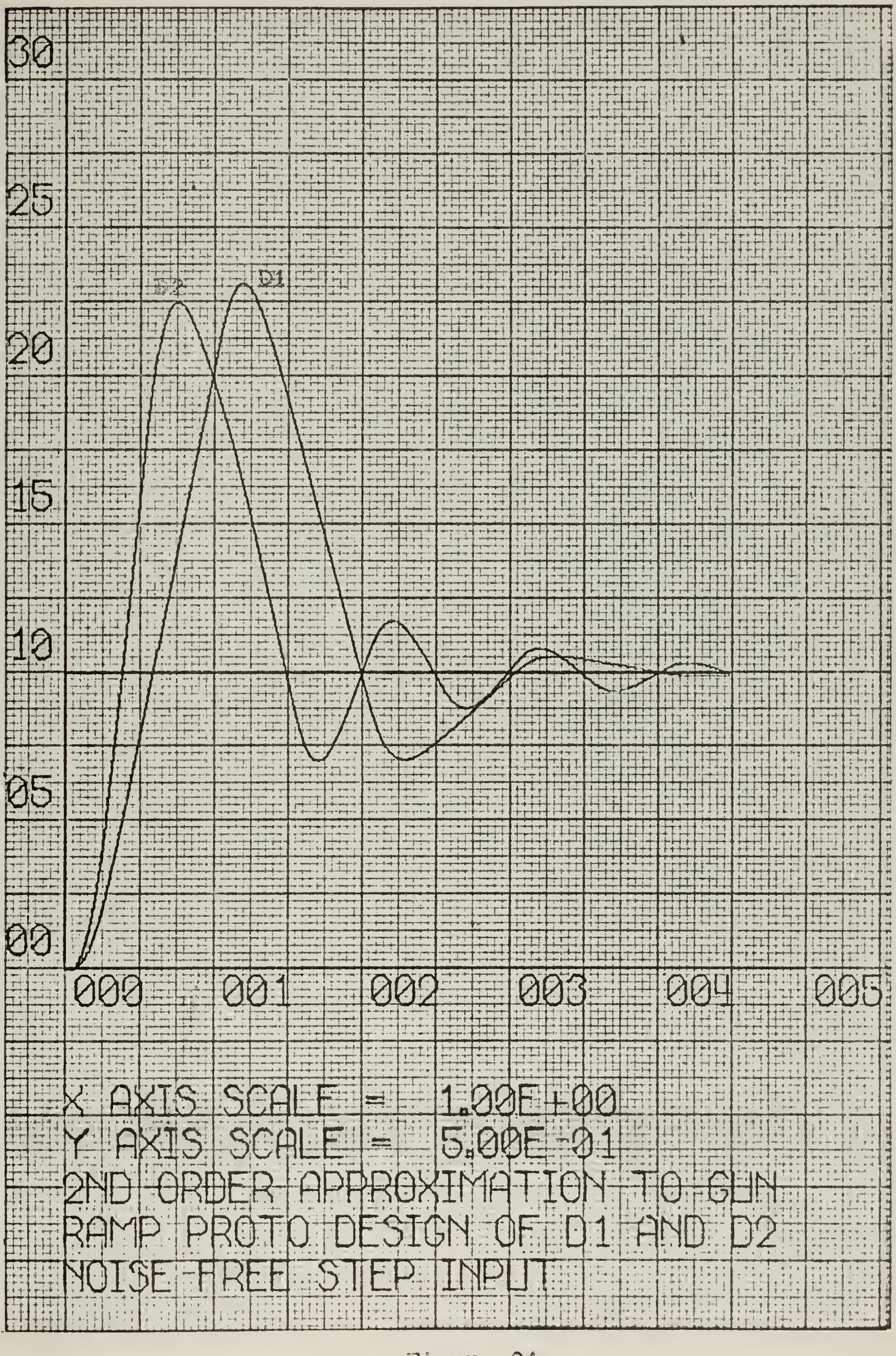

30
25
20
15
10
05
00
D2
D1
000
001
002
003
004
005
X AXIS SCALE = 1.00E+00
Y AXIS SCALE = 5.00E-01
2ND ORDER APPROXIMATION TO GUN
RAMP PROTO DESIGN OF D1 AND D2
NOISE FREE STEP INPUT

Figure 24

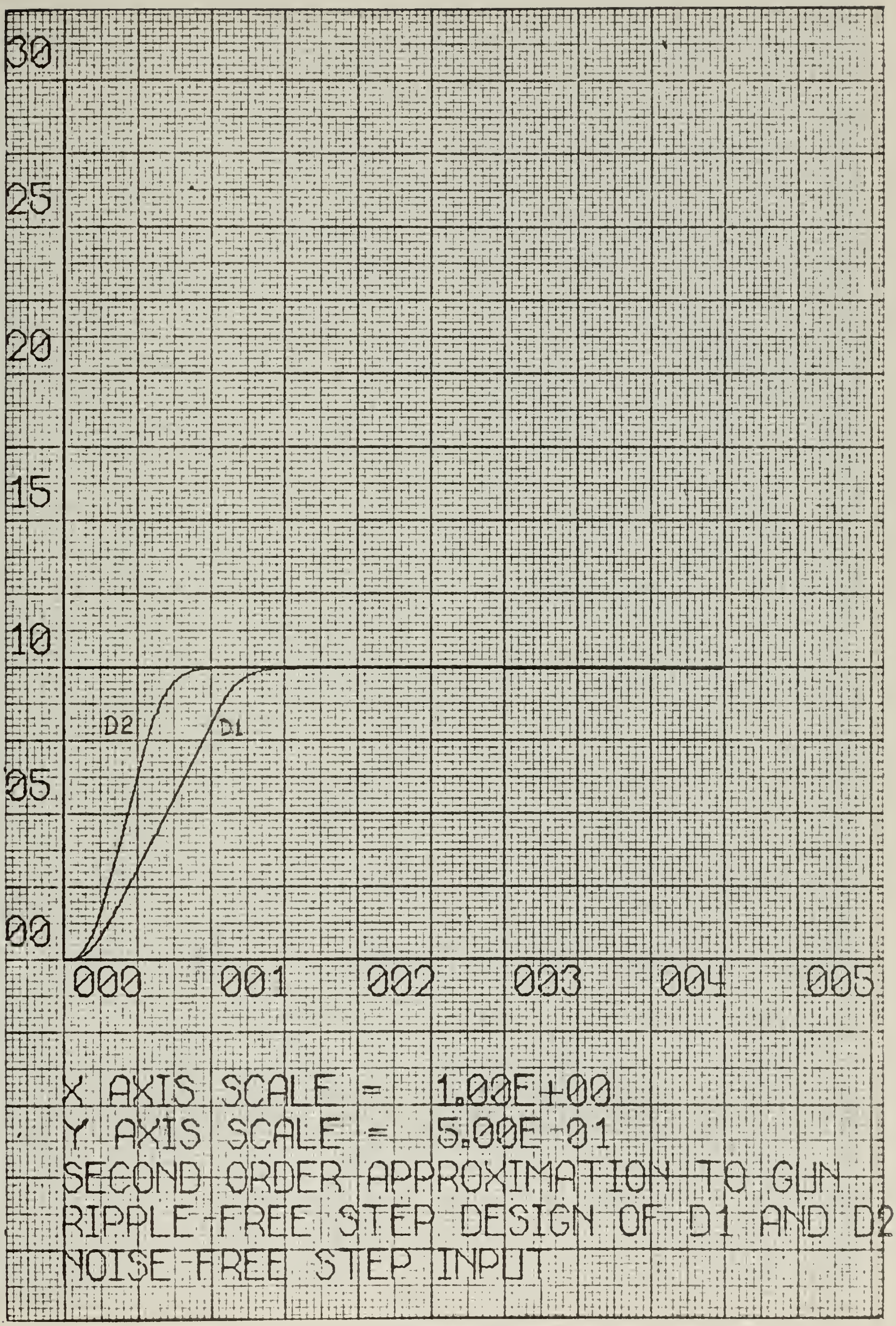
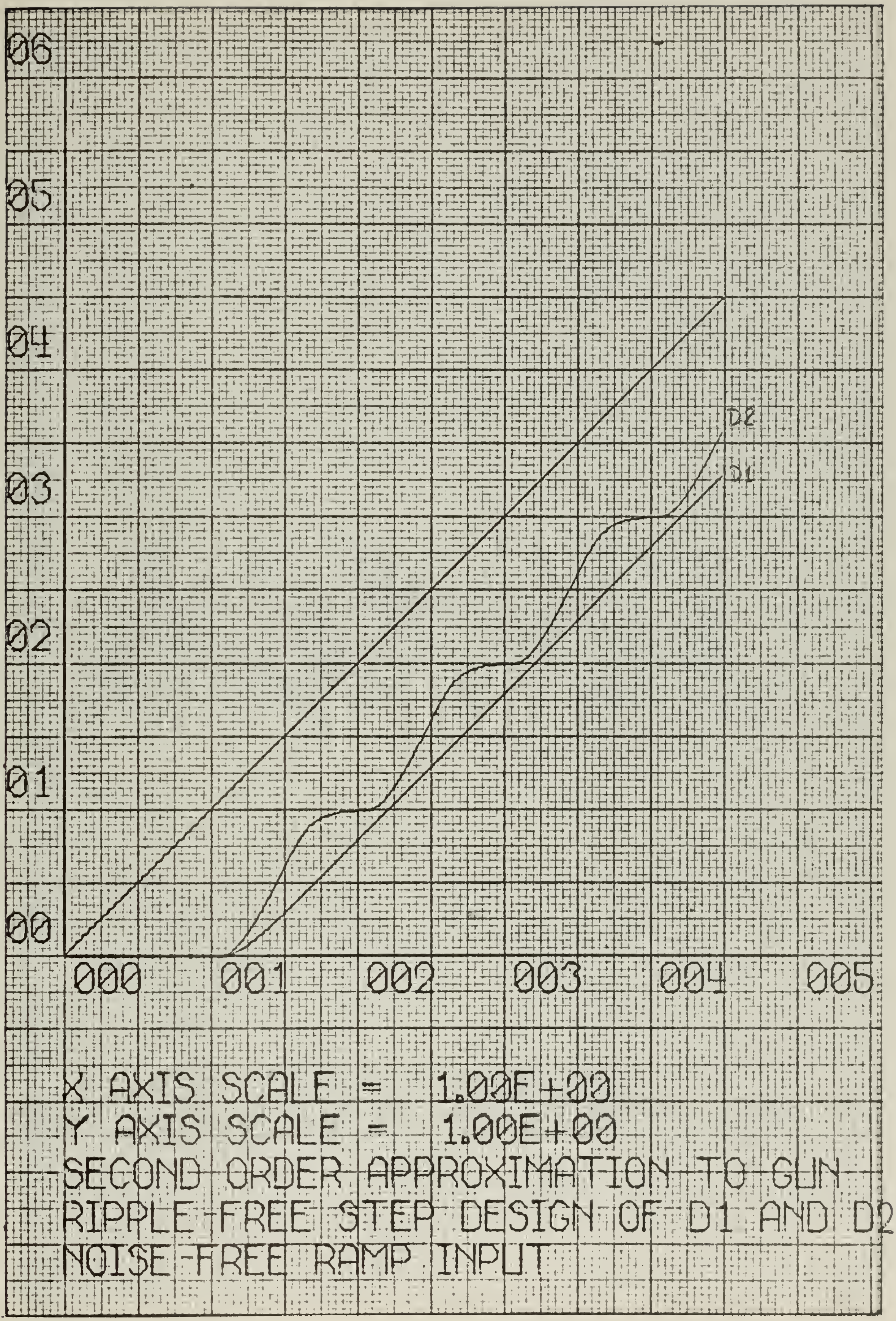

30
25
20
15
10
05
00
D2
D1
000
001
002
003
004
005
X AXIS SCALE = 1.00E+00
Y AXIS SCALE = 5.00E-01
SECOND ORDER APPROXIMATION TO GUN
RIPPLE FREE STEP DESIGN OF D1 AND D2
NOISE FREE STEP INPUT

Figure 25



06
05
04
03
02
01
00
000
001
002
003
004
005
D2
D1
X AXIS SCALE = 1.00E+00
Y AXIS SCALE = 1.00E+00
SECOND ORDER APPROXIMATION TO GUN
RIPPLE-FREE STEP DESIGN OF D1 AND D2
NOISE-FREE RAMP INPUT

Figure 26

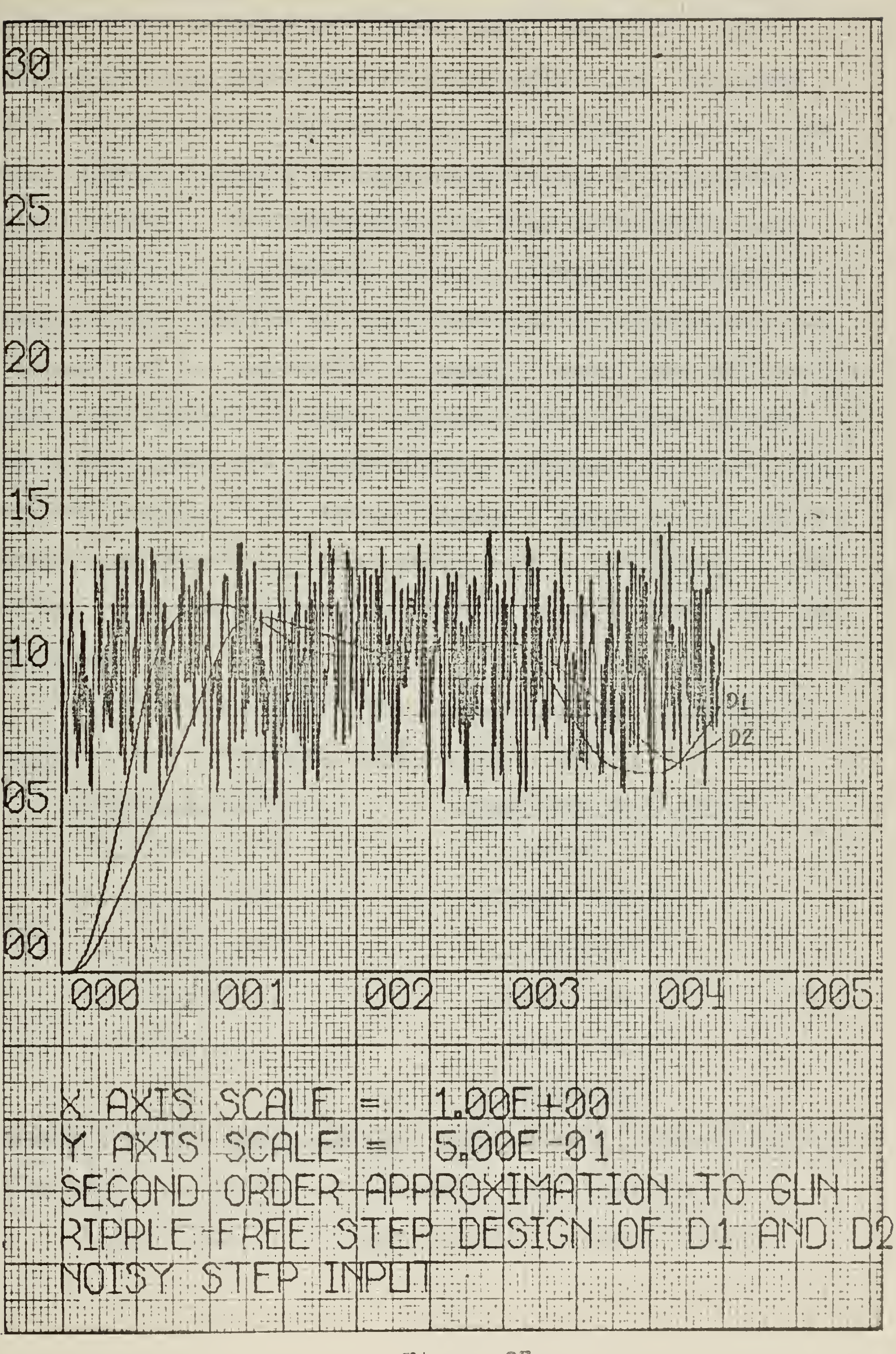

30
25
20
15
10
05
00
000
001
002
003
004
005
D1
D2
X AXIS SCALE = 1.00E+00
Y AXIS SCALE = 5.00E-01
SECOND ORDER APPROXIMATION TO GUN
RIPPLE FREE STEP DESIGN OF D1 AND D2
NOISY STEP INPUT

Figure 27

The simplicity of the design techniques for sampling rates of one or two samples per second which result in valid simplifying approximations is quite appealing, but there lurks an intuitive suspicion that these rates are too slow for a system required to accurately follow a high-speed aerial target. This suspicion has been confirmed in private communications between John B. Slaughter and Gene F. Franklin, the results of which are that the damped natural frequency of the gun mount in train is about one and one-half cycles per second and that sampling should occur between six and twelve times the damped natural frequency of any system for good results. For these reasons a sampling rate of ten samples per second has been decided on by the group at NEL, and the remaining simulation studies of this work will use this rate.

For $T = 0.1$ seconds the pulse transfer function of the plant and zero-order hold is shown below:

Numerator of G(z)	Denominator of G(z)
$21.91241531z^{-1}$	1.00000000
$131.76730855z^{-2}$	$-1.61606197z^{-1}$
$63.94180401z^{-3}$	$.72871398z^{-2}$
$3.08436589z^{-4}$	$-.11877672z^{-3}$
$.00419048z^{-5}$	$.00612622z^{-4}$
	$-.00000152z^{-5}$

The roots of the numerator are located at the origin and at -.00139908, -.05265621, -.47316625, and -5.48614151

The presence of the zero of G(z) outside the unit

circle complicates the design effort slightly as it did in the third order plant considered earlier. The same methods apply and the resulting pulse transfer function of the process which gives prototype response to a step input is $K(z) = .154174866z^{-1} + .845825134z^{-2}$

Figure 28 shows this process responding to a noise-free step input, and Figure 29 shows the same process with a noise-free velocity input. The remarks made earlier concerning prototype processes apply and will not be repeated here. Figure 30 shows the results obtained with this design when the step input contains additive noise. The best that can be said about the plant's behaviour in this case is that it remained stable and in the vicinity of the input signal. Not so for a noisy ramp input, to which the response of the sampled data system was unstable. It appears that the sensitivity of a sampled data system to noise increases with the order of the input signal.

The prototype design for zero steady state error in response to a velocity input begins with the equations below:

$$K(z) = (1 + 5.4861415z^{-1})(a_1z^{-1} + a_2z^{-2})$$

$$1 - K(z) = (1 - z^{-1})^2(1 + b_1z^{-1})$$

From these are obtained three equations to be solved for three unknowns, namely

$$a_1 = 2 - b_1,\ -5.4861415a_2 = b_1, \text{ and}$$

$$a_2 - 5.4861415a_1 = 2b_1 - 1$$

Solving these, the resulting over-all process is

$$K(z) = .438754756z^{-1} + 2.122490488z^{-2} - 1.561245244z^{-3}$$

The coefficients of this process as well as the coefficients of the numerator and denominator of $G(z)$ and the value of the zero outside the unit circle were supplied to the simulator program, which then calculated a single-rate digital controller. The responses of the system to a noise-free ramp and a noise-free step are shown in Figures 31 and 32 respectively. With the addition of noise, instability again resulted when the input was a ramp, while for a noisy step the output was at least bounded for the duration of the experiment, although it began to look inauspicious towards the end. See Figure 33. In an attempt to achieve at least a semblance of stability in response to a noisy ramp signal the noise deviation was reduced from 0.2 to .05, but to no avail. A future investigation will be made into the stability threshold for noisy ramp signals.

It was then decided to design for prototype step response using a staleness factor of 0.5. The design steps are shown below:

Prototype $K(z) = .154174866z^{-1} + .845825134z^{-2}$ as found earlier. This function contains the zero of $G(z)$ located outside the unit circle at $z = 5.48614151$. The process pulse transfer function with a 0.5 staleness factor is given by $K(z) = \dfrac{a(0.154174866z^{-1} + 0.845825134z^{-2})}{1 - 0.5z^{-1}}$

For a unit step input, application of the Final Value Theorem shows that "a" must have the value 0.5. Thus $K(z)$ is:

$$K(z) = \frac{.077087433\, z^{-1} + 0.422912567\, z^{-2}}{1 - 0.5 z^{-1}}$$

Substituting in $D(z) = \frac{Q(z)}{P(z)} \cdot \frac{K(z)}{1 - K(z)}$

where P(z) is the numerator of G(z) given on page 71, and

Q(z) is the denominator of G(z) given on page 71,

and cancelling the common factors $(1-z^{-1})(1+5.48614151z^{-1})$,

the single-rate digital controller is given by

Numerator of D(z)	Denominator of D(z)
.003517980	1.00000000
$-.002167290z^{-1}$	$.95013414z^{-1}$
$.000396307z^{-2}$	$.24861928z^{-2}$
$-.000021510z^{-3}$	$.01088370z^{-3}$
$.000000025z^{-4}$	$.00001039z^{-4}$

This pulse transfer function was supplied to the simulator program; test inputs of noise-free step, noise-free ramp, noise-free acceleration, and noisy step were applied with the results shown in Figures 34 through 37 inclusive. The main point of interest in these results is that step response in the presence of noise for this staleness factor design is perhaps the best-behaved so far, although it is not good by any means. In addition, noisy inputs of ramp and acceleration were applied, but both of these gave unstable responses as in previous designs.

The final design effort to be described in this work was directed toward achieving ripple-free response to a velocity input when sampling at a rate of ten samples per second. For this design all of the zeros of G(z) must be contained in K(z), so we begin by writing

$$K(z) = (1+5.48614151z^{-1})(1+.47316625z^{-1})(1+.05265621z^{-1})$$
$$(1+.00139908z^{-1})(a_1z^{-1} + a_2z^{-2})$$

Since we are designing for zero steady state error for a ramp input, then it must be true that

$$1-K(z) = (1-z^{-1})^2(1 + b_1z^{-1} + b_2z^{-2} + b_3z^{-3} + b_4z^{-4})$$

After multiplying out and equating like coefficients of z we obtain six simultaneous equation in six unknowns which are presented below in matrix form.

$$\begin{bmatrix} 1.0 & 0 & 1.0 & 0 & 0 & 0 \\ 6.013363 & 1.0 & -2.0 & 1.0 & 0 & 0 \\ 2.918063 & 6.013363 & 1.0 & -2.0 & 1.0 & 0 \\ .140759 & 2.918063 & 0 & 1.0 & -2.0 & 1.0 \\ .000191 & .140759 & 0 & 0 & 1.0 & -2.0 \\ 0 & .000191 & 0 & 0 & 0 & 1.0 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} = \begin{bmatrix} 2.0 \\ -1.0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

This matrix equation was solved by computer using a Gaussian elimination library routine. With the results obtained therefrom, the coefficients of the process, K(z), were calculated as follows:

$$\begin{aligned} A(1) &= 2 - b_1 &&= .319530814 \\ A(2) &= 2b_1 - 1 - b_2 &&= 1.70120542 \\ A(3) &= 2b_2 - b_1 - b_3 &&= -.39202846 \\ A(4) &= 2b_3 - b_2 - b_4 &&= -.59772472 \\ A(5) &= 2b_4 - b_3 &&= -.03094094 \\ A(6) &= -b_4 &&= -.00004212 \end{aligned}$$

These process coefficients were supplied to the simulator program along with the coefficients of P(z) and Q(z) and the values of the four plant zeros to be included. The program then calculated D(z) for the single-rate sampled data system. For a test input of a noise-free ramp the output has no overshoot or inter-sample ripple and reaches

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

zero error in 0.4 seconds. See Figure 38. The response to a noise-free step is shown in Figure 39, where the "tuned" nature of the design is again exhibited. However, after the initial large overshoot the step response, too, is essentially ripple free and steady state in 0.4 seconds.

For the noisy ramp signal shown in Figure 40 the sampled data system was unstable, while for a step signal with the same amount of additive noise the response can be seen in Figure 41. Note that the time scale in Figure 41 is five times longer than in the noise-free experiments, and the response to the noisy signal shows no signs of settling down any further at the end of five seconds than it did at the end of the first second. Clearly, noise of this power cannot be tolerated in the input signal to a sampled data system regardless of the digital controller design.

[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]
[illegible]

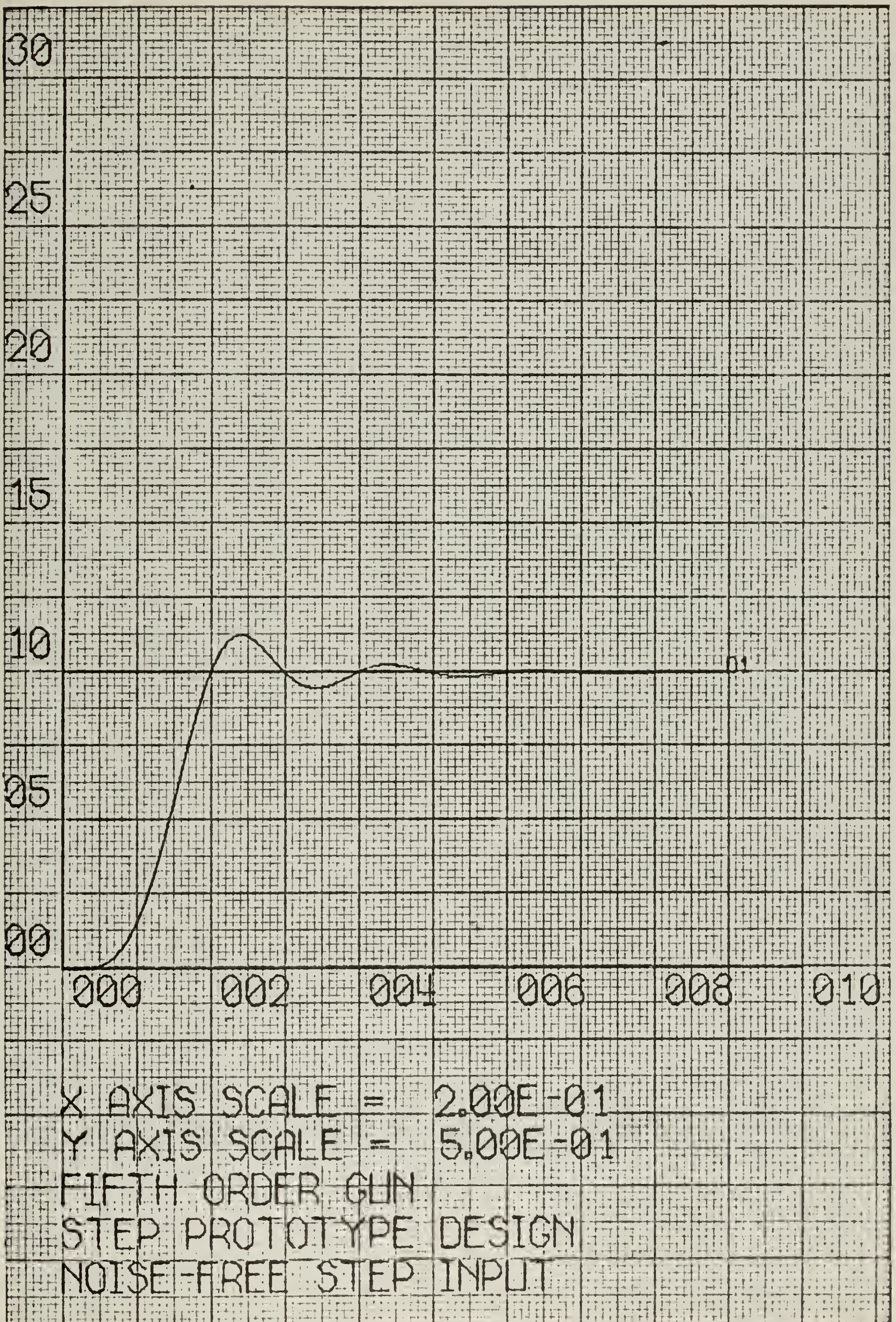

30
25
20
15
10
05
00
000
002
004
006
008
010
X AXIS SCALE = 2.00E-01
Y AXIS SCALE = 5.00E-01
FIFTH ORDER GUN
STEP PROTOTYPE DESIGN
NOISE-FREE STEP INPUT

Figure 28

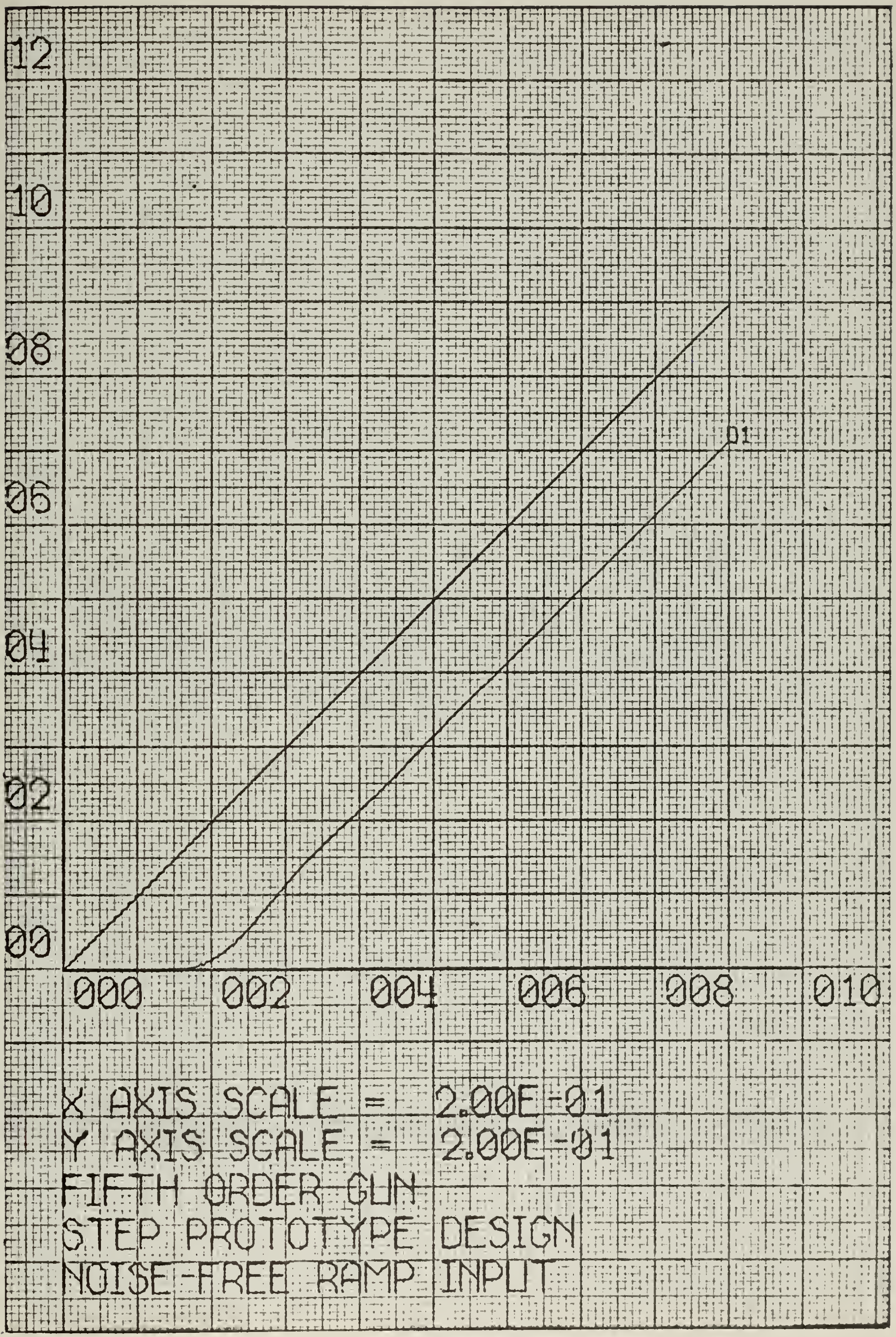

12
10
08
06
04
02
00
01
000
002
004
006
008
010
X AXIS SCALE = 2.00E-01
Y AXIS SCALE = 2.00E-01
FIFTH ORDER GUN
STEP PROTOTYPE DESIGN
NOISE-FREE RAMP INPUT

Figure 29

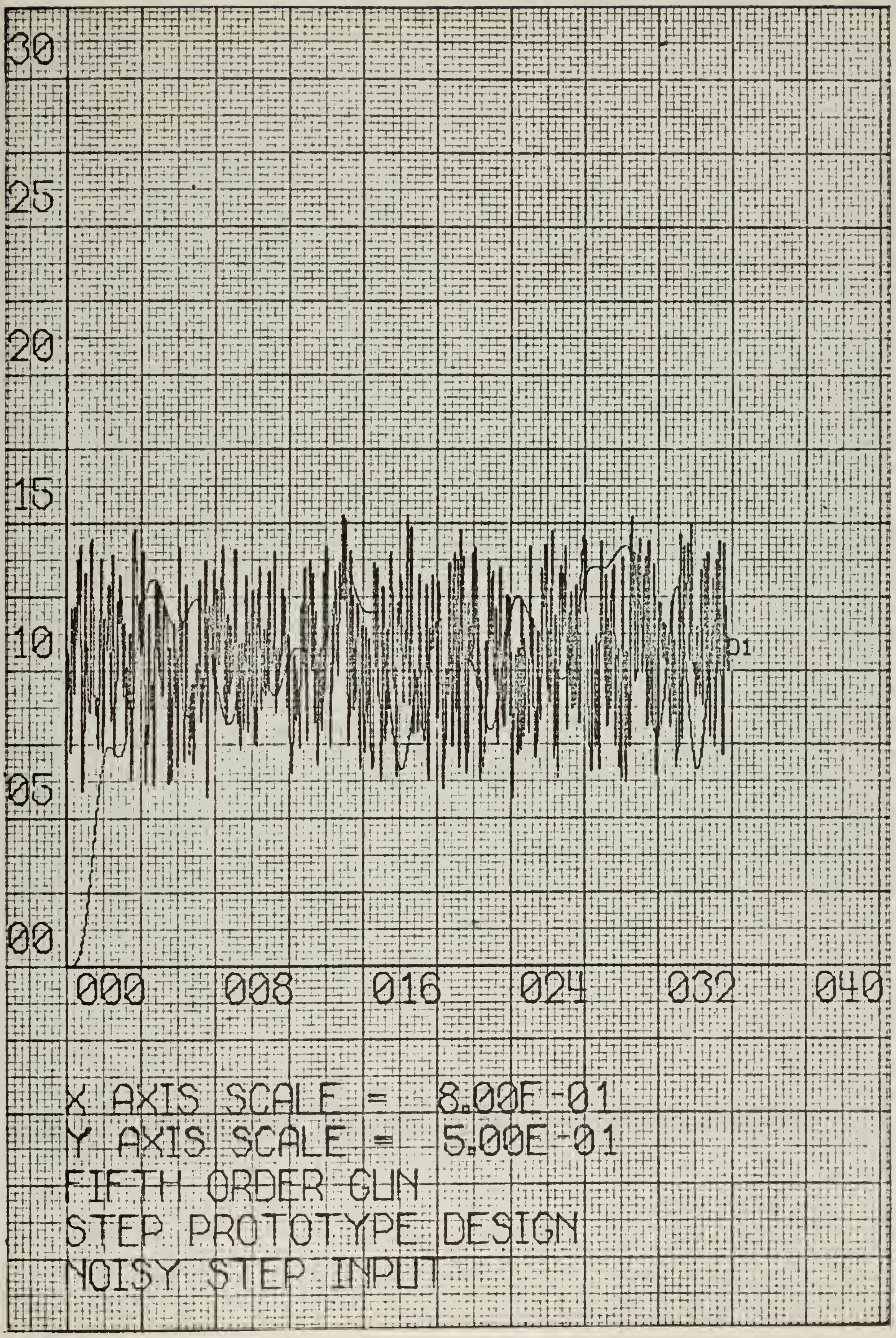

30
25
20
15
10
05
00
000
008
016
024
032
040
D1
X AXIS SCALE = 8.00E-01
Y AXIS SCALE = 5.00E-01
FIFTH ORDER GUN
STEP PROTOTYPE DESIGN
NOISY STEP INPUT

Figure 30

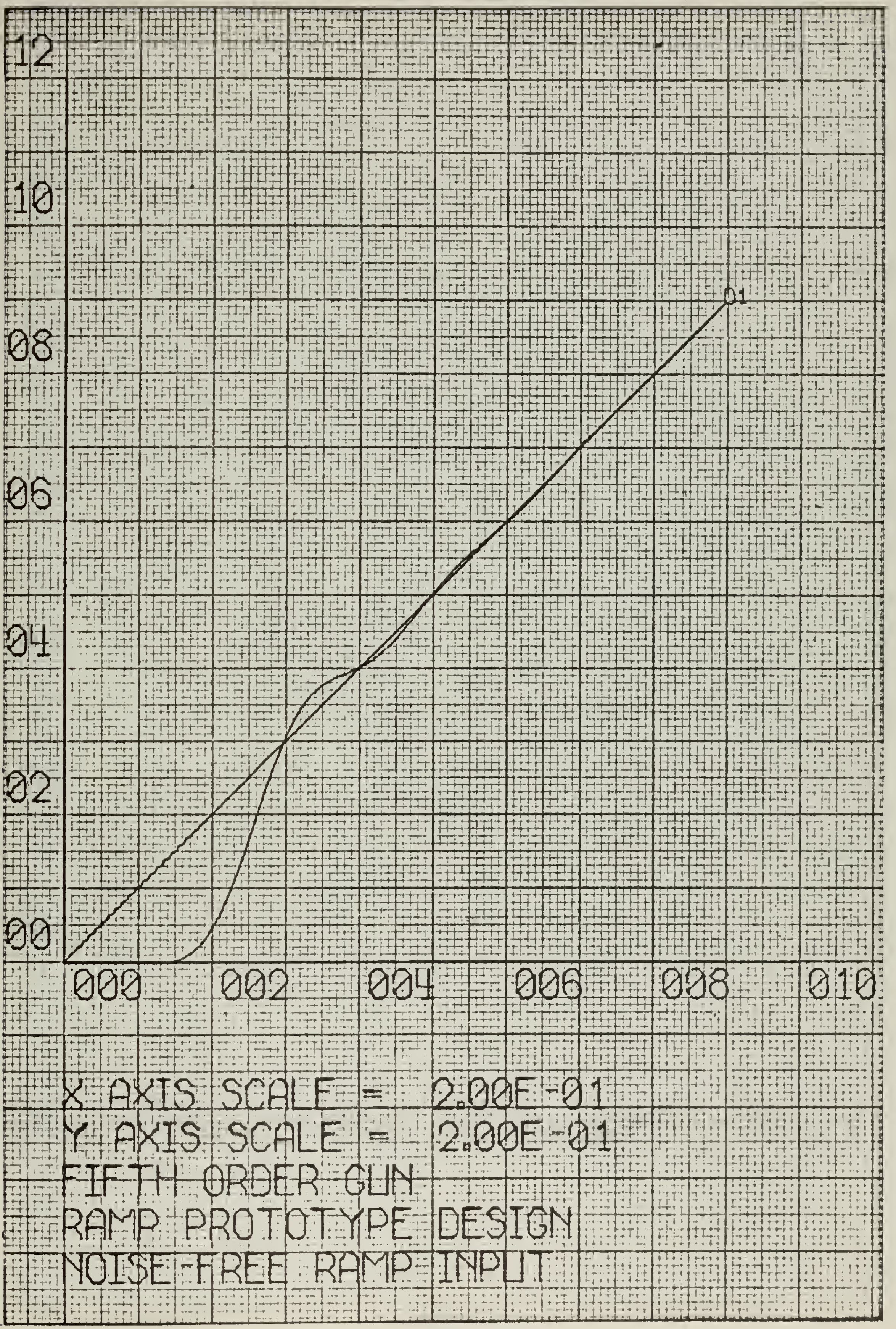

12
10
08
06
04
02
00
01
000
002
004
006
008
010
X AXIS SCALE = 2.00E-01
Y AXIS SCALE = 2.00E-01
FIFTH ORDER GUN
RAMP PROTOTYPE DESIGN
NOISE-FREE RAMP INPUT

Figure 31

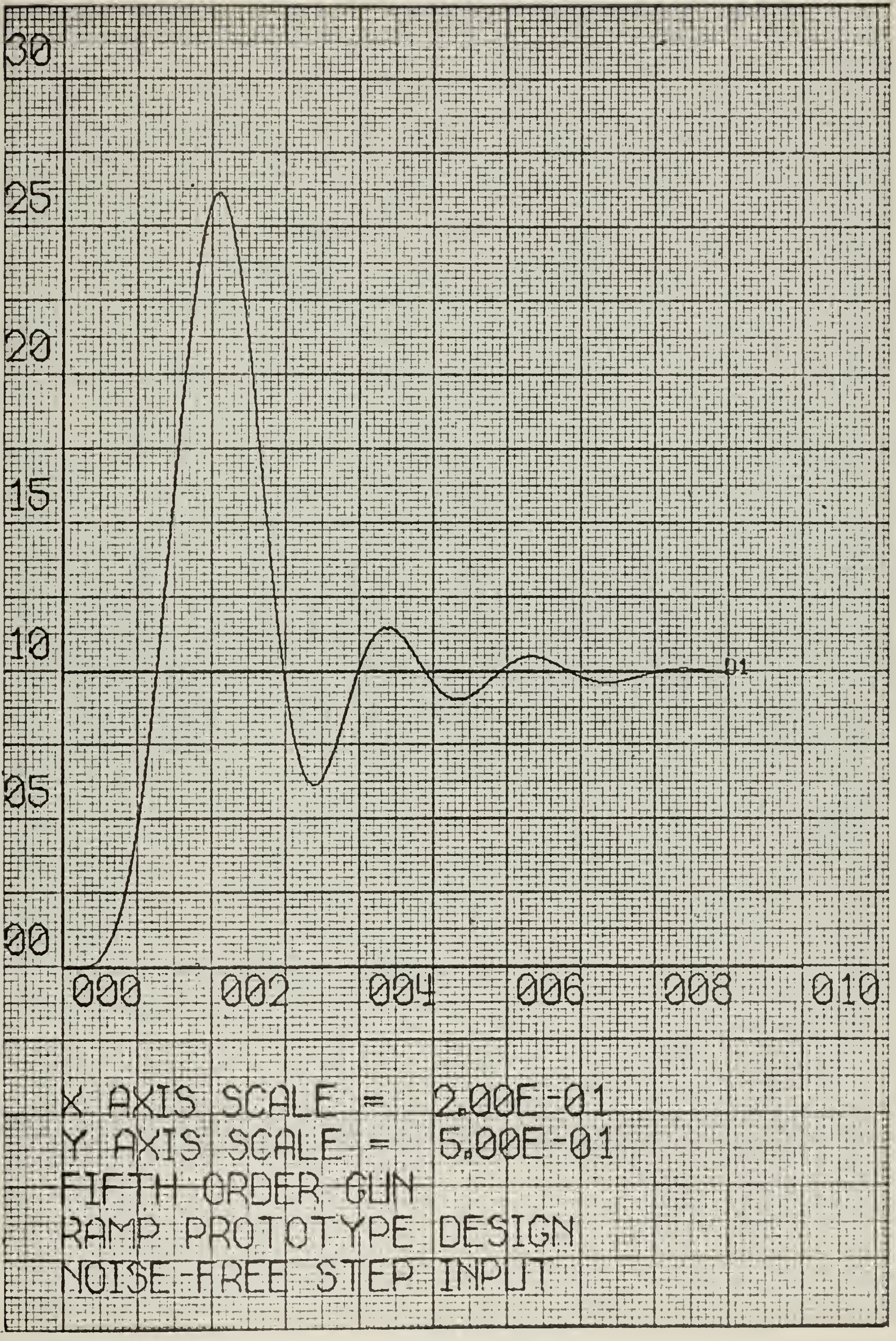

30
25
20
15
10
05
00
01
000
002
004
006
008
010
X AXIS SCALE = 2.00E-01
Y AXIS SCALE = 5.00E-01
FIFTH ORDER GUN
RAMP PROTOTYPE DESIGN
NOISE-FREE STEP INPUT

Figure 32

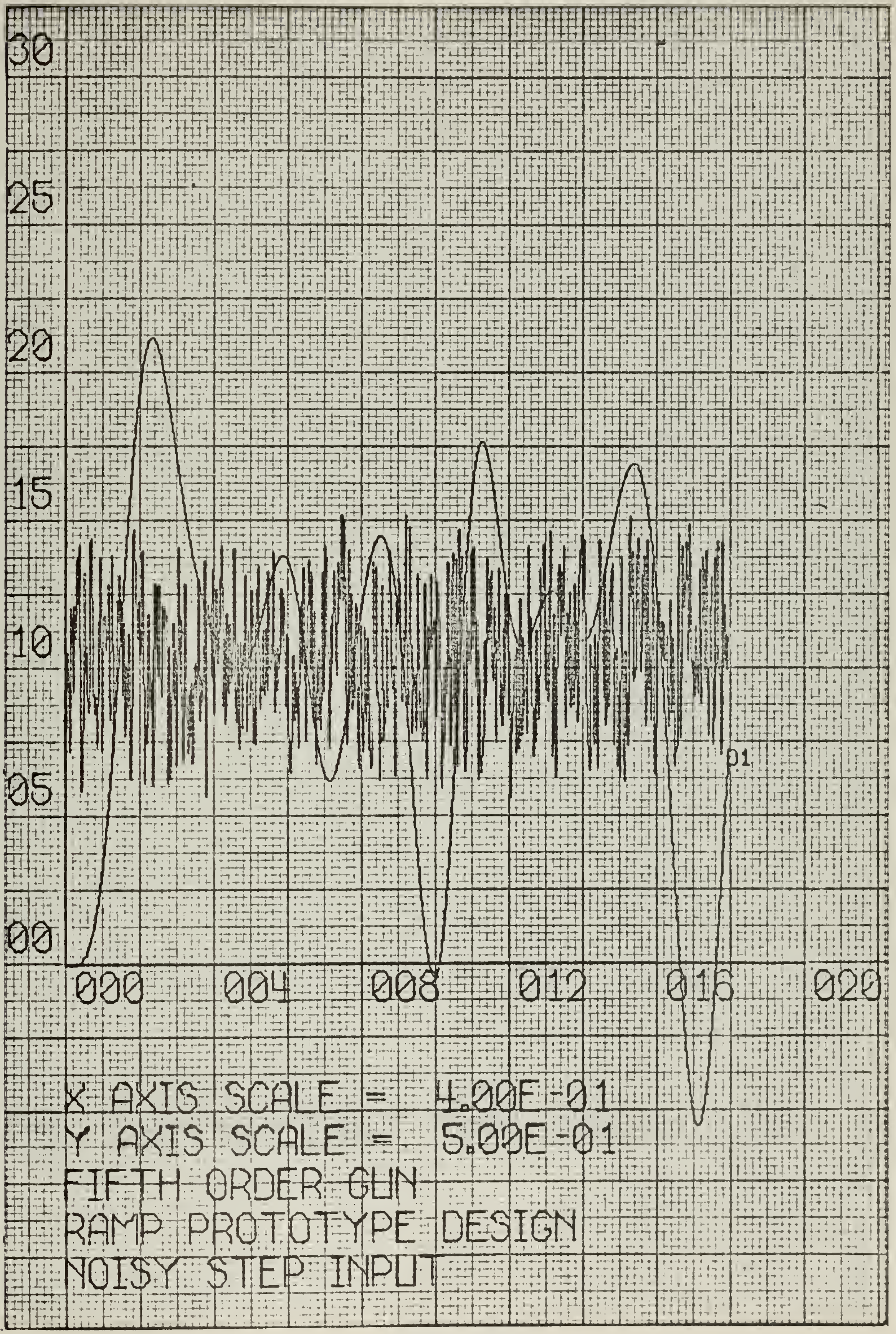

30
25
20
15
10
05
00
000
004
008
012
016
020
01
X AXIS SCALE = 4.00E-01
Y AXIS SCALE = 5.00E-01
FIFTH ORDER GUN
RAMP PROTOTYPE DESIGN
NOISY STEP INPUT

Figure 33

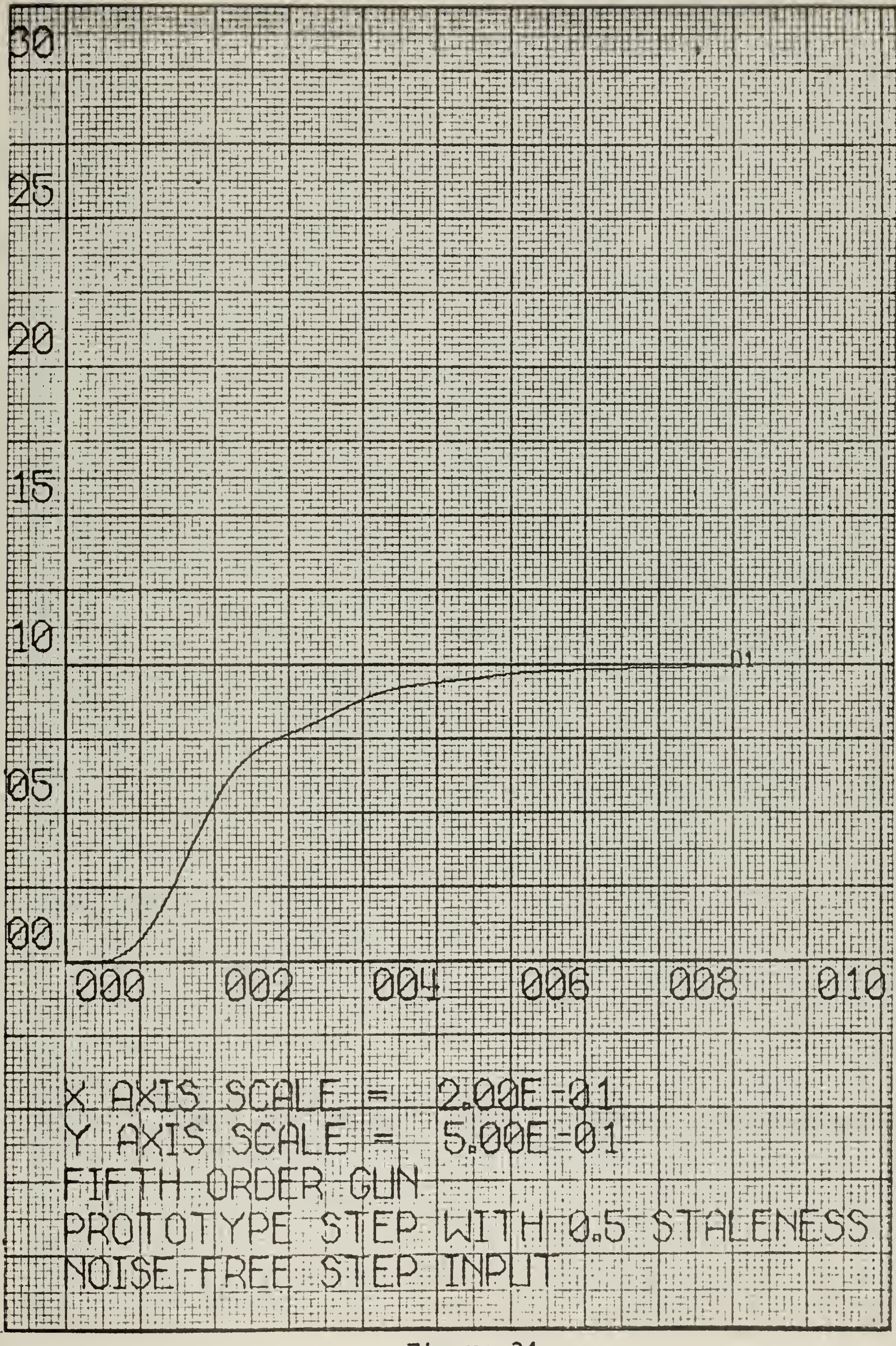

30
25
20
15
10
05
00
01
000
002
004
006
008
010
X AXIS SCALE = 2.00E-01
Y AXIS SCALE = 5.00E-01
FIFTH ORDER GUN
PROTOTYPE STEP WITH 0.5 STALENESS
NOISE-FREE STEP INPUT

Figure 34

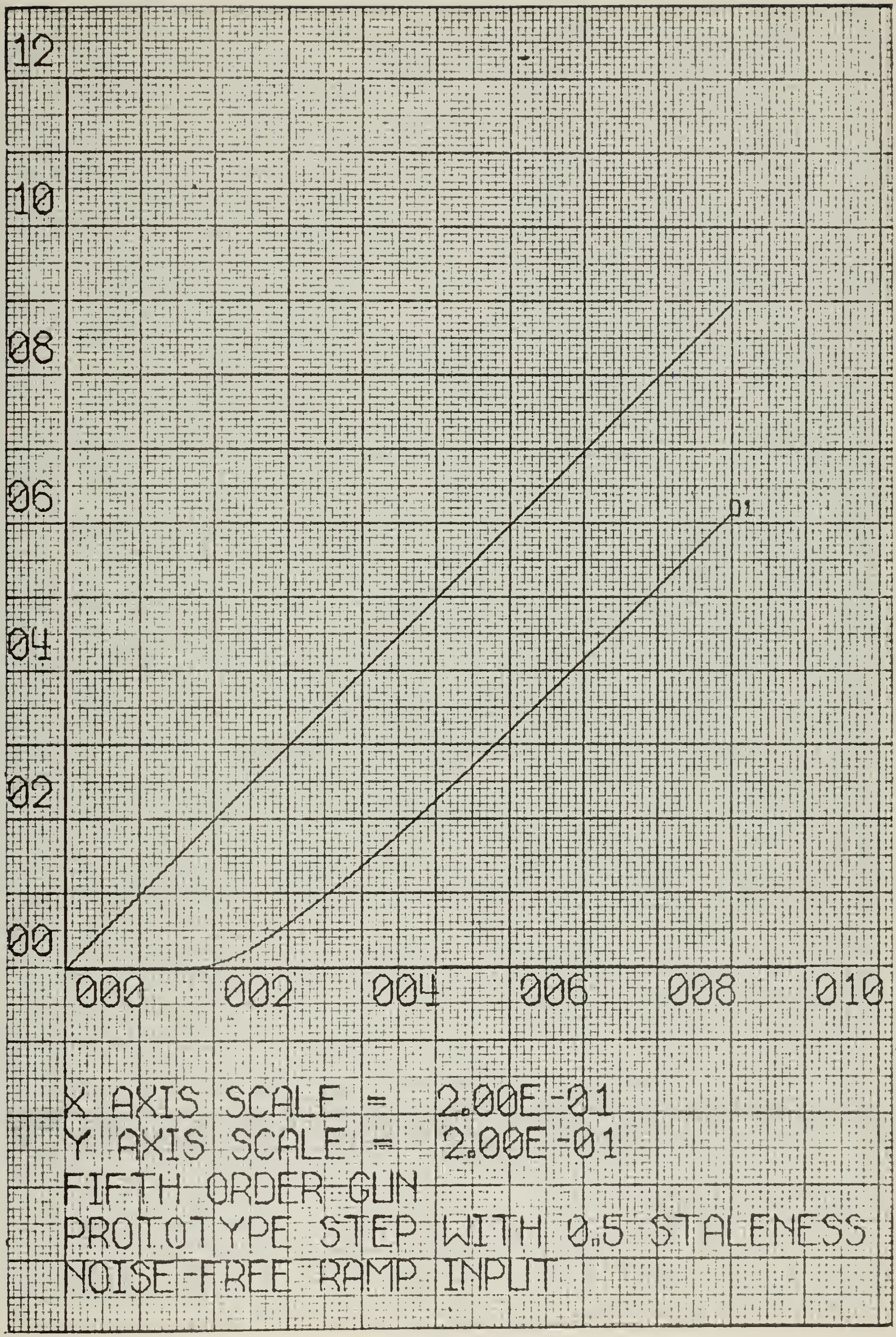

12
10
08
06
04
02
00
01
000
002
004
006
008
010
X AXIS SCALE = 2.00E-01
Y AXIS SCALE = 2.00E-01
FIFTH ORDER GUN
PROTOTYPE STEP WITH 0.5 STALENESS
NOISE-FREE RAMP INPUT

Figure 35

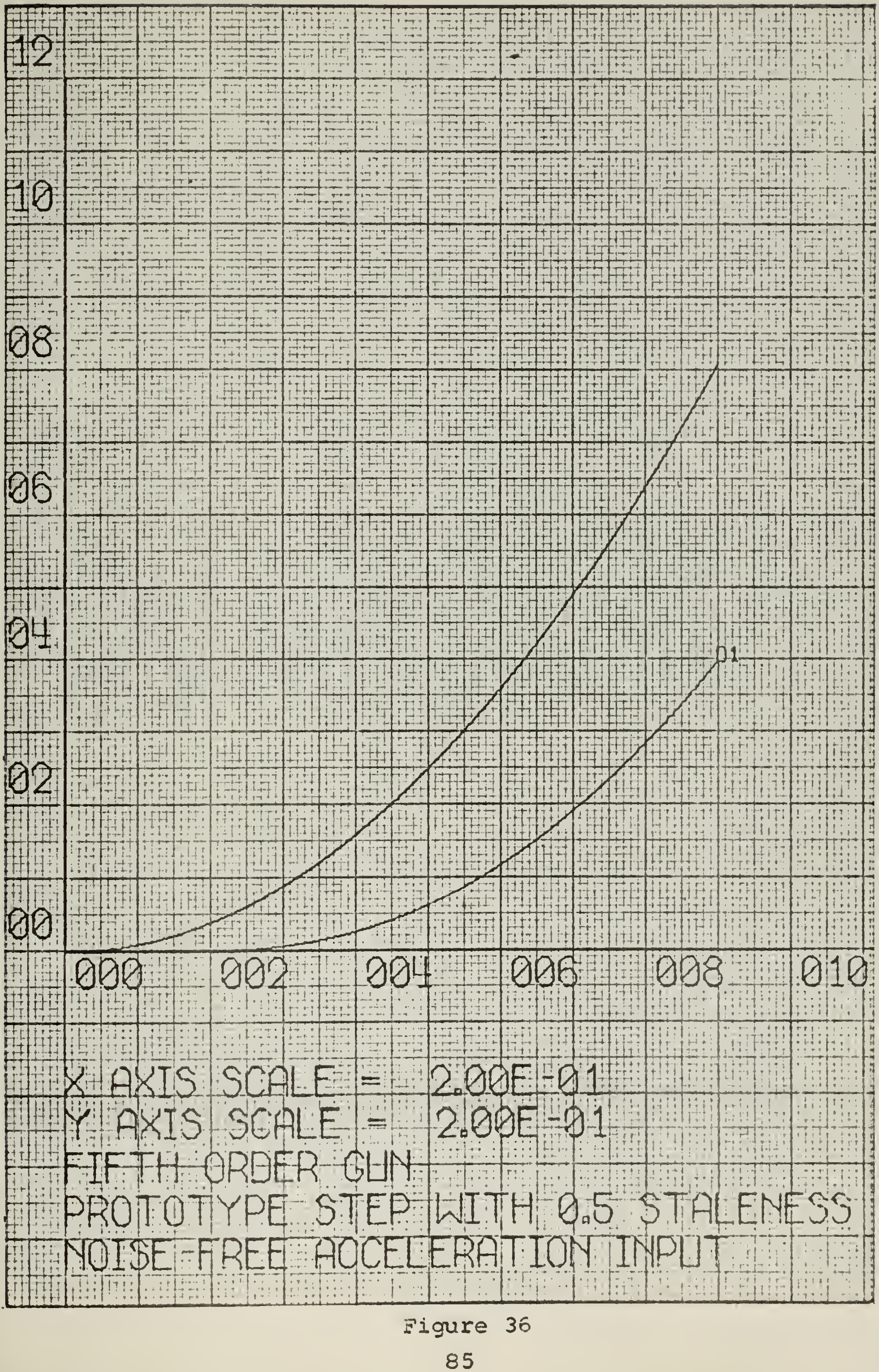

12
10
08
06
04
02
00
000
002
004
006
008
010
01
X AXIS SCALE = 2.00E-01
Y AXIS SCALE = 2.00E-01
FIFTH ORDER GUN
PROTOTYPE STEP WITH 0.5 STALENESS
NOISE-FREE ACCELERATION INPUT

Figure 36

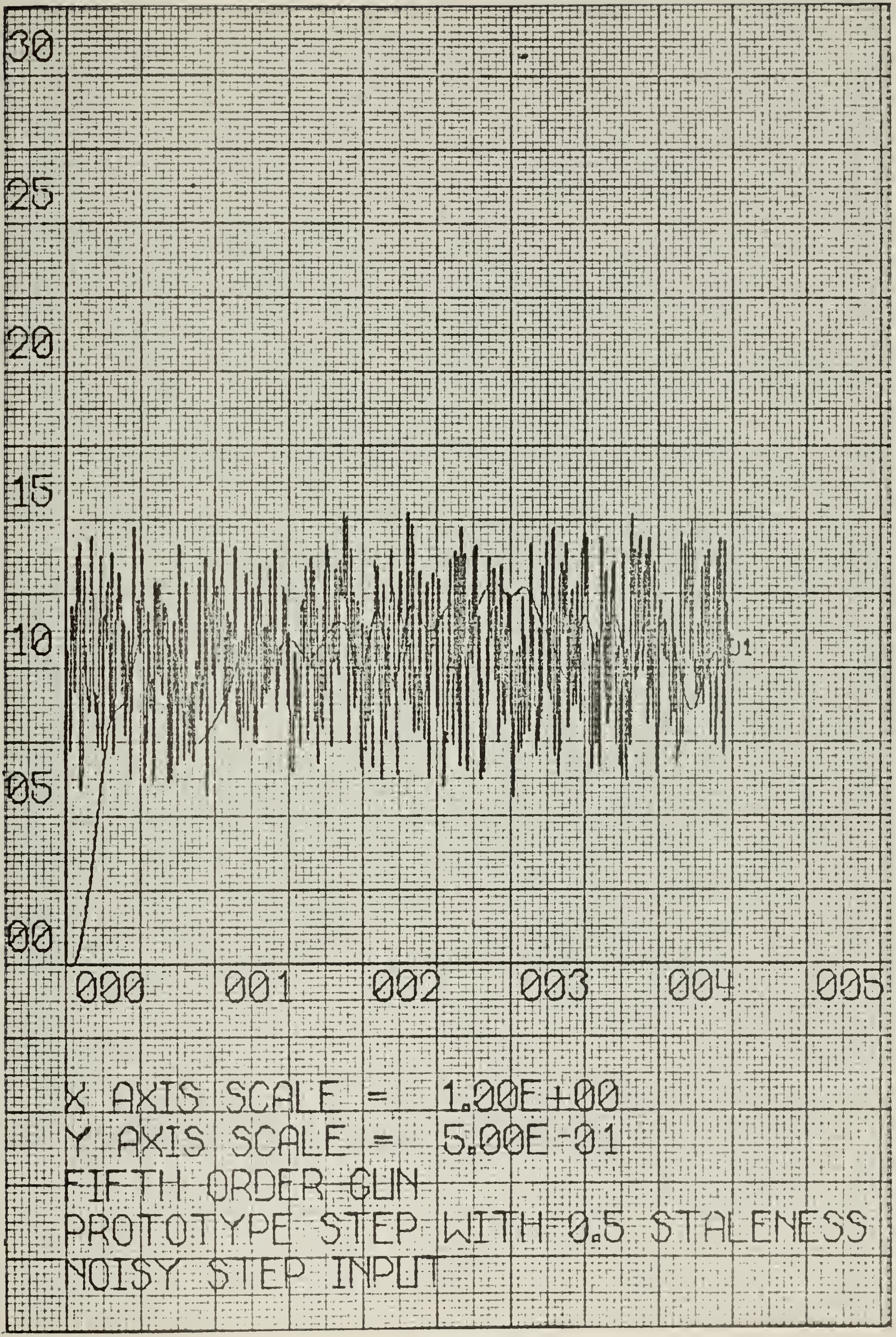

30
25
20
15
10
05
00
000
001
002
003
004
005
X AXIS SCALE = 1.00E+00
Y AXIS SCALE = 5.00E-01
FIFTH ORDER GUN
PROTOTYPE STEP WITH 0.5 STALENESS
NOISY STEP INPUT

Figure 37

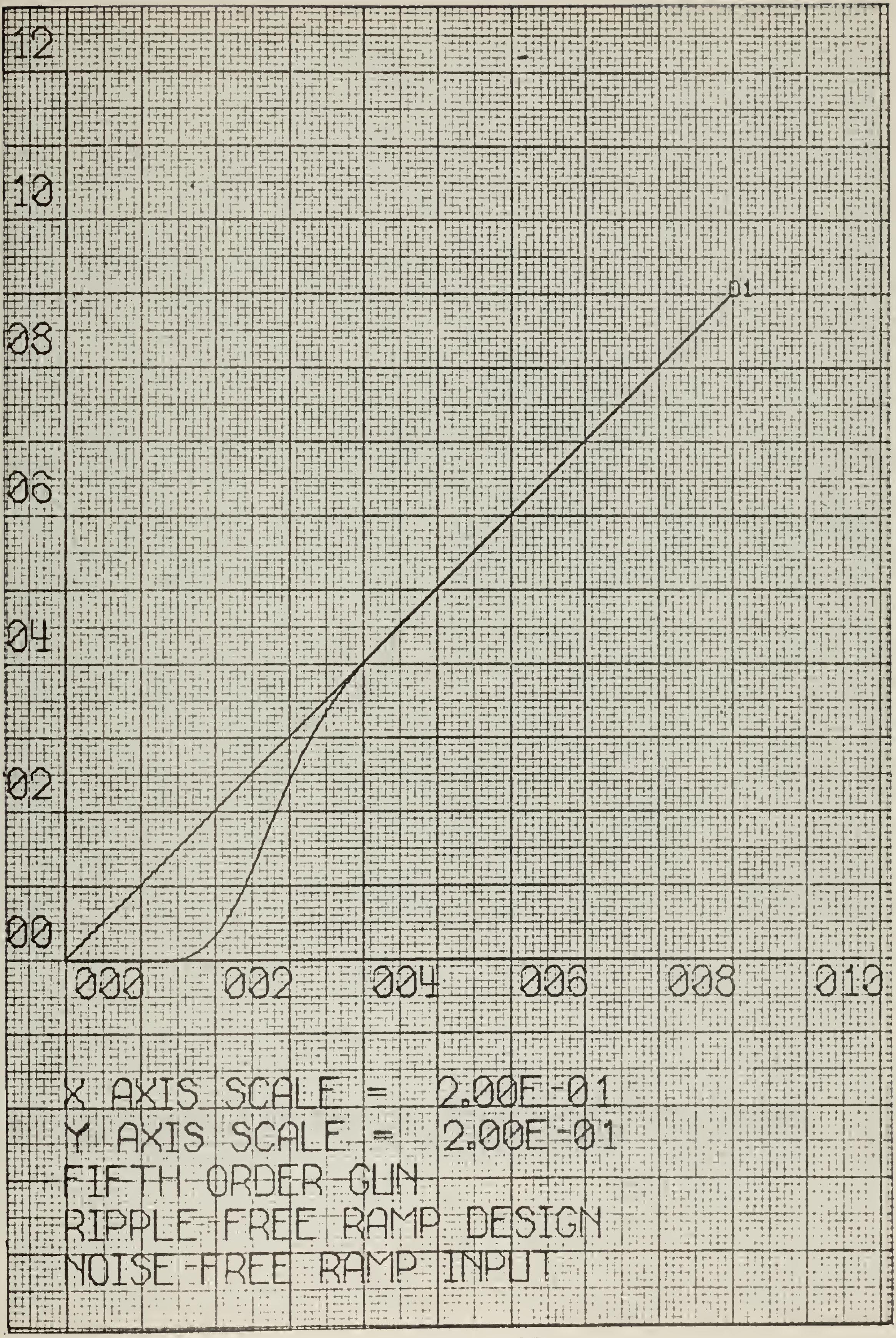

12
10
08
06
04
02
00
01
000
002
004
006
008
010
X AXIS SCALE = 2.00E-01
Y AXIS SCALE = 2.00E-01
FIFTH ORDER GUN
RIPPLE FREE RAMP DESIGN
NOISE FREE RAMP INPUT

Figure 38

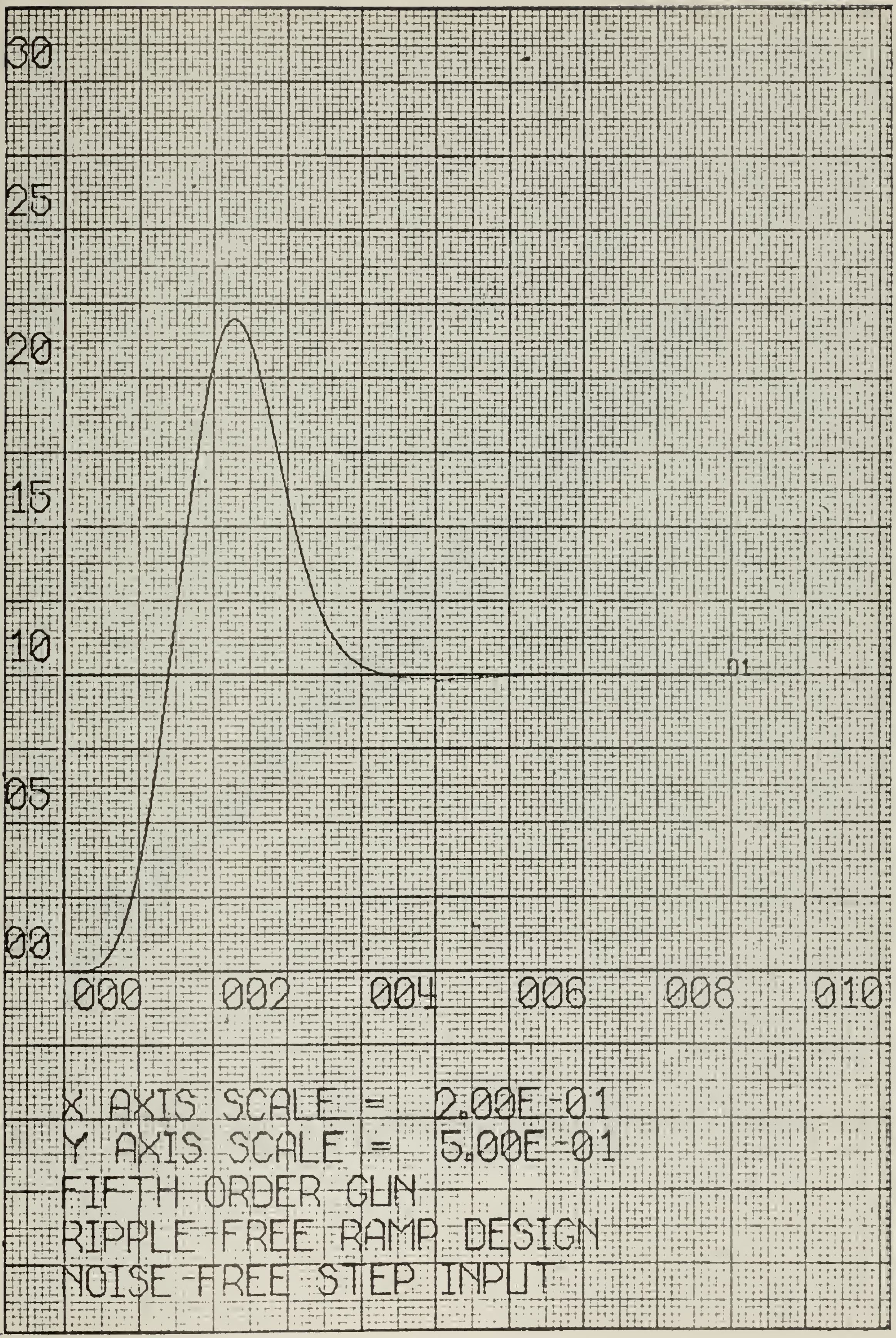

30
25
20
15
10
05
00
.01
000
002
004
006
008
010
X AXIS SCALE = 2.00E-01
Y AXIS SCALE = 5.00E-01
FIFTH ORDER GUN
RIPPLE FREE RAMP DESIGN
NOISE FREE STEP INPUT

Figure 39

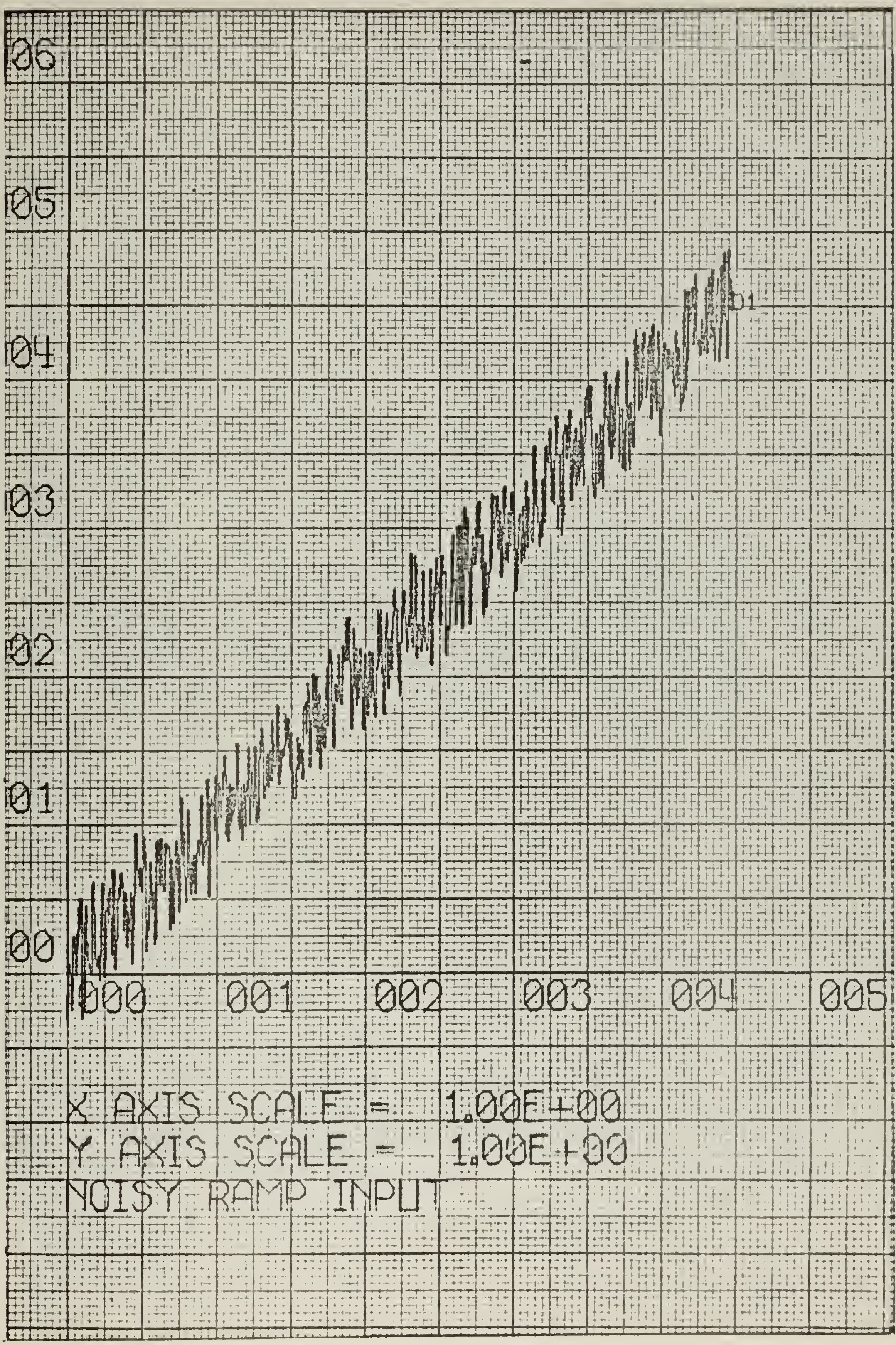

06
05
04
03
02
01
00
000
001
002
003
004
005
X AXIS SCALE = 1.00E+00
Y AXIS SCALE = 1.00E+00
NOISY RAMP INPUT

Figure 40

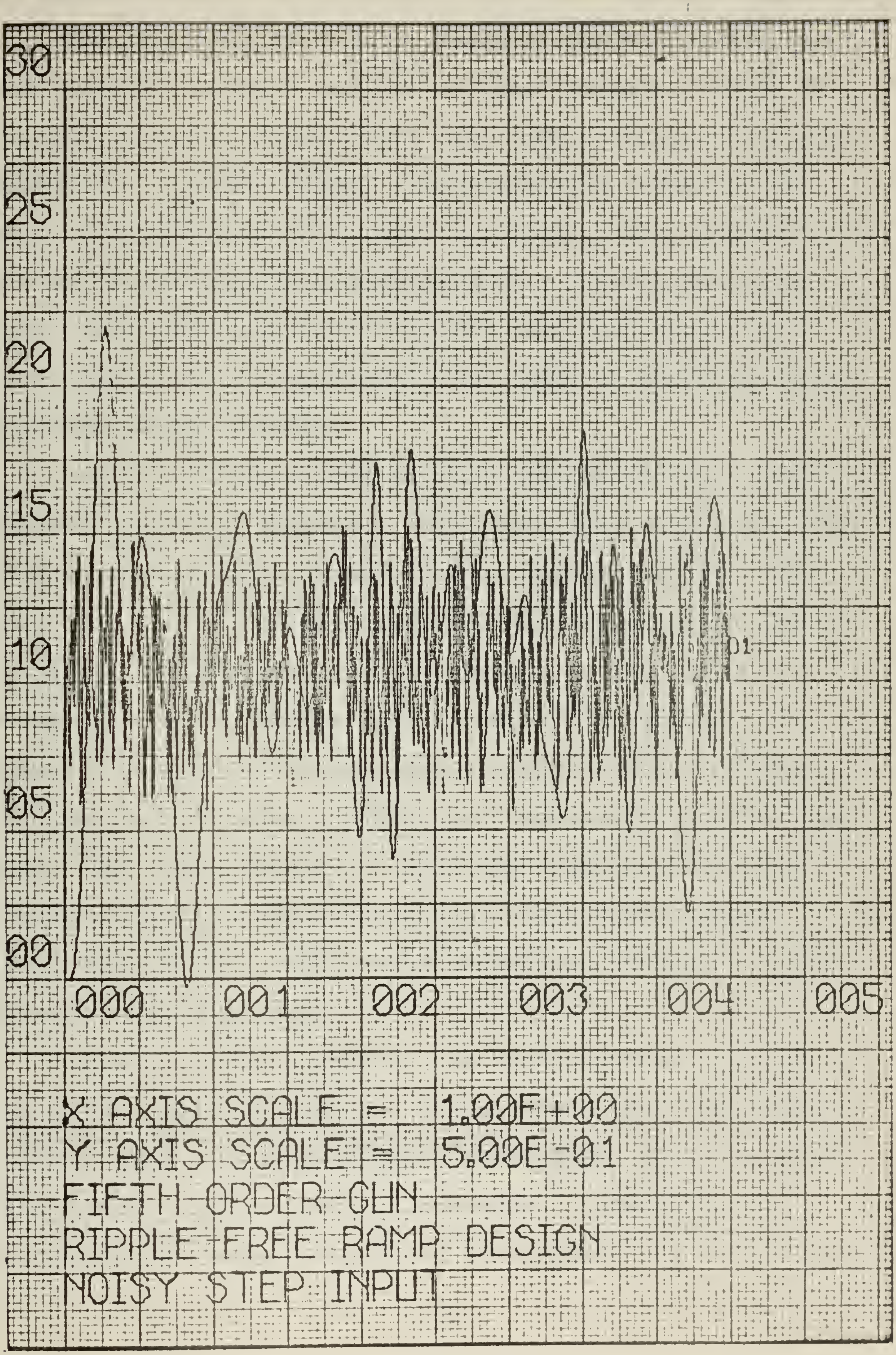

30
25
20
15
10
05
00
000
001
002
003
004
005
X AXIS SCALE = 1.00E+00
Y AXIS SCALE = 5.00E-01
FIFTH ORDER GUN
RIPPLE FREE RAMP DESIGN
NOISY STEP INPUT

Figure 41

8. Conclusions.

The behaviour of a unity feedback control system has been simulated using a digital computer program. Using this simulator program studies have been made of digital control methods for various plants, with particular attention focused on the control of a Naval gun mount. The azimuth function of the gun mount is described by a fifth order differential equation, and it is believed that the conclusions pertaining to it are applicable, at least in part, to other shipboard weapon systems.

The use of multi-rate digital controllers was investigated. In every instance, the double-rate controller provided faster response by one-half a sampling period. However, only in the special case of ripple-free design for a step input was the double-rate response an unalloyed improvement over the single-rate response. In general, transient control and inter-sample ripple was not noticeably improved by using a double-rate controller for the gun mount. For other plants of different frequency response, advantage might be taken of the fact that inter-sample ripple when using a double-rate controller occurs at twice the error-sampling rate.

Pulse transfer functions (or Z-transforms) are a function of the sampling period. It was found that for certain sampling rates valid simplifying assumptions could be made which facilitated the design procedure. Care should be exercised that these simplifying assumptions are not made on

the basis of a sampling rate which is too slow in relation to the damped natural frequency of the system concerned. In this regard the approximations made to the pulse transfer function of the gun mount were inappropriate since they were based on a sampling rate roughly equal to the damped natural frequency instead of six to twelve times greater as must be the case for adequate control.

Extreme sensitivity of sampled data systems to noise in the input signal was demonstrated. No satisfactory method of coping with input noise by design of the over-all process was found. So-called "variance reduction" processes showed little merit as the result of these simulation studies. The design criteria which came closest to producing satisfactory response to a noisy step input was the prototype with staleness factor of 0.5.

Without compensation, either analog or digital, the gun mount is closed-loop unstable. Several digital control functions were developed which gave stable response to noise-free input signals. None of these gave stable operation when noise was present in a first order (ramp) input or a second order (acceleration) input, but for a noisy zero order (step) input digital control achieved a response that was at least bounded for the duration of the experiment, albeit unsatisfactory for most conceivable applications. It is concluded that sampled data system sensitivity to additive noise in the input increases with the order of the input signal.

The computer program described in Appendix II for

finding Z-transforms was very helpful in this investigation. It is hoped that this program, as well as the control system simulator upon the results of which this work was based, will prove useful to others in the field of sampled data systems. It is further hoped that this work has, in a small way, indicated the truthfulness of the assertion that digital compensation techniques introduce a degree of flexibility and accuracy in a control system that is not presently achievable with state-of-the-art analog techniques.

BIBLIOGRAPHY

1. Alfred J. Monroe, Digital Processes for Sampled Data Systems, John Wiley and Sons, Inc., New York, 1962.

2. Julius T. Tou, Digital and Sampled Data Control Systems, McGraw-Hill Book Company, New York, 1959.

3. E. I. Jury, Sampled Data Control Systems, John Wiley and Sons, Inc., New York, 1959.

4. John R. Ragazzini and Gene F. Franklin, Sampled Data Control Systems, McGraw-Hill Book Company, New York, 1958.

5. J. B. Slaughter, W. J. Dejka, and W. V. Kershow, Digital Sampled Data Feedback Systems for Shipboard Control Applications, Progress Report, Navy Electronics Laboratory Report 1125, 9 July 1962.

6. B. B. Newell, A Proposal for an Analog-Digital-Analog System for Use With the CDC 1604 Digital Computer, Digital Control Laboratory Report, USNPGS, Monterey, California, 28 December 1961.

7. R. H. Barker, The Pulse Transfer Function and Its Application to Sampling Servo Systems, Proc. IEE, part IV, Monograph 43, 15 July 1952.

8. J. E. Bertram, Factors in the Design of Digital Controllers for Sampled Data Feedback Control Systems, Trans. AIEE, paper 56-209, 1956.

9. G. M. Kranc, Compensation of an Error-Sampled System by a Multi-Rate Controller, Trans. AIEE, part II, pp. 149-159, 1957.

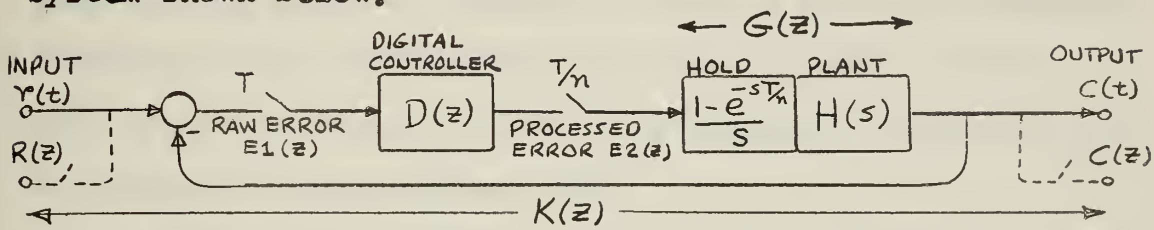
10. Modern Computing Methods, National Physical Laboratory, Teddington, England, p. 57. Philosophical Library, Inc., New York, 1961.

APPENDIX I

DIGITAL COMPUTER PROGRAM FOR SIMULATION OF A SAMPLED DATA SYSTEM

General Description of Program ALSTAPP

This program is a simulation of the hybrid control system shown below.



Three choices are available for the test input; namely, a unit step, a unit ramp, and a unit acceleration. The plant may be up to thirtieth order and is currently subject to the following restrictions:

(1) No plant zeros are allowed, i.e., the numerator of H(s) must be a constant.

(2) The plant may not have a pole at the origin higher than first order, i.e., limited to Type 0 and Type 1 servos.

(3) Only the constant term may be missing in the differential equation describing the plant.

In essence, the plant is restricted to having a transfer function of the general form

$$H(s) = \frac{VK}{S(S+P_1)(S+P_2)\cdots(S+P_{K-1})} \quad \text{simple poles, Pi}$$

or

$$H(s) = \frac{VK}{A(K+1)s^K + A(K)s^{K-1} + \cdots + A(2)s + A(1)}$$

where A(1) is normally zero for Type 1 servo and is the only missing term allowed.

The rate at which the raw error, E1, is sampled can be selected at will. This sample period T is chosen by the user and supplied to the computer.

DELT is the small increment of time used in the simulation of the continuous plant by numerical solution of the plant's differential equation. This is done one thousand times, so that the total problem time is 1000 x DELT. DELT should be chosen as small as possible consistent with the desired total problem time.

Although one thousand values of the solution are generated and are available for use by a computer plotting routine, only every tenth solution is produced on the Analex printer, since it was felt that a column of one hundred values is sufficient for perusal by eye and more than adequate for plotting by hand.

This program performs four major tasks. The first three are performed every time the program is executed, but it is optional with the user whether or not the fourth task will be done. These tasks are:

(1) Simulation of a continuous, un-compensated, unity feedback system for the given plant.

(2) Calculation of coefficients for a single-rate digital controller.

(3) Simulation of the sampled-data system employing a single-rate digital controller using the coefficients calculated above.

(4) Simulation, if desired, of a sampled-data system

employing a double-rate controller. The coefficients of the double-rate controller must be supplied by the user, and if they are not, the fourth task is not performed.

Information must be supplied to the program via data cards. Detailed instructions for the preparation and contents of these cards is given later. Seventeen (17) cards are required if the program is to do all four tasks, while only the first thirteen (13) need be supplied if simulation of the double rate controller is not desired. CAUTION: The cards must be presented to the computer in <u>exactly</u> the order and form specified. The slightest error will cause faulty operation or, perhaps, no operation at all.

<u>Discussion of Information Supplied by the User</u>

The general nature of the information that must be supplied to the computer is described next.

(1) The user must choose the small time increment DELT to be used in the numerical method solution of the plant's differential equation. The smaller this increment the more accurate the solution. However, the total time simulated will be only one thousand times DELT, so the user should first determine the time span of interest...i.e., an estimate of how long it will take the system to achieve steady state, or how much of the transient condition it is desired to observe for lightly damped systems...then divide this total time by 1000 to obtain the value for DELT.

(2) The user must choose the sampling period to be employed in the sampled-data system with single-rate

controller simulation. This value is called T.

(3) The user chooses the test input to be applied to the system. Unit values of step, ramp, and acceleration are available.

(NOTE 1: These first three pieces of information, namely, DELT, T, and INPUT are supplied by data card number one.)

(4) The specifications of the continuous plant are supplied to the computer by data cards 2, 3, and 4. The order of the plant is denoted by K. This is the highest power of the Laplace variable, S, in the denominator of the plant transfer function, H(s). The numerator gain constant of H(s) is denoted by VK. The order of the pole at the origin is denoted by NTYPE and is limited to 0 or 1. These values, K, VK, and NTYPE are supplied by data card number two.

(5) H(s) must be known in the polynomial form, that is to say, a numerator gain constant, VK, over a denominator polynomial in S. The constant coefficients of this denominator polynomial are denoted by A(i) and must be supplied to the computer. The only coefficient which may have a zero value is the constant term A(1). The number of coefficients is denoted by NA and is supplied by data card number 3. Even though A(1) is normally zero (in a Type 1 servo) it must be included in the count, NA, on card 3 and its zero value shown in the first ten columns of data card number four (4). In the usual case of a Type 1 servo, the value of NA is one more than the order, K, of the plant.

(6) For the simulation of the sampled-data systems, Z-transform techniques are implemented. Data reconstruction is accomplished by a simulated zero order hold, which must be included in the Z-transform approach to the plant. If we call the plant and hold-circuit combination G(s), then

$$G(s) = \frac{1-e^{-sT}}{s} \cdot H(s)$$

and the pulse transfer function (or z-transform) is

$$G(z) = (1-z^{-1})\, \mathcal{Z}\, \frac{H(s)}{s} = \frac{P(z)}{Q(z)}$$

This pulse transfer function for the plant and hold must be supplied by the user. Its general form is a ratio of polynomials in Z^{-1} as shown below:

$$G(z) = \frac{P(z)}{Q(z)} = \frac{P(1)z^{-1} + P(2)z^{-2} + \cdots \; P(NP)z^{-NP}}{Q(1) + Q(2)z^{-1} + Q(3)z^{-2} + \cdots + Q(NQ)z^{-NQ+1}}$$

where NP is the number of terms (and also the highest power of Z^{-1}) in the numerator, P(Z), and NQ is the number of terms in the denominator, Q(Z). In order for G(Z) to be a physically realizable pulse transfer function, the term Q(1) must be a constant, and in fact, is usually unity.

(7) Although G(Z) is supplied to the computer in polynomial form, it must be known in factored form, in order that any poles or zeros of G(Z) that lie on or outside the unit circle in the Z-plane may be accounted for. Usually, the plant is open loop stable and only zeros need be accounted for, but the program will also process information and provide the digital controller coefficients when the plant has a pole outside the unit circle. All such zeros must be supplied by the user under the generic

term ZERO(i) and the number of them is denoted by NZ. Up to ninety-nine poles and/or zeros may be accounted for.

(8) The overall pulse transfer function of the entire system is denoted by K(Z) and is defined as

$$K(z) = \frac{C(z)}{R(z)}$$

For the purposes of this program, K(Z) must be a numerator polynomial only. This implies a "finite-memory" process. If an infinite-memory or "non-finite settling time process" is desired, then it must be approximated by a finite number of terms of the infinite series obtained by long division. The program will operate on up to ninety-nine (99) such terms for K(Z). The general form for K(Z) is

$$K(z) = AK(1)\,z^{-1} + AK(2)\,z^{-2} + \cdots + AK(NK)\,z^{-NK}$$

where NK is the number of terms provided. K(Z) must be supplied by the user.

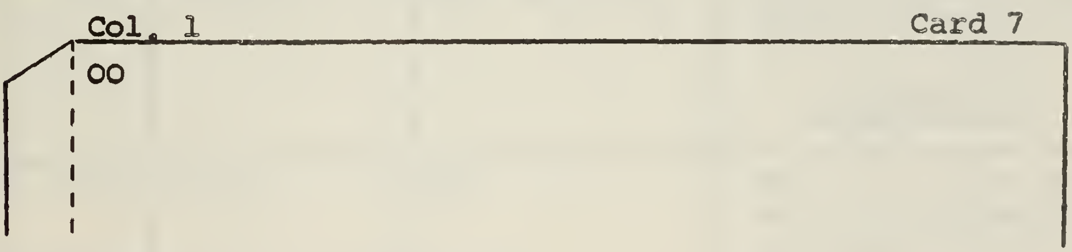
<u>NOTE 2</u>: In order to supply an array of numerical data to the computer two (2) data cards are required. The first card tells the computer how many values are in the array, and the next card (or cards) supplies these values. Assuming for the moment that no array contains more than eight (8) values, the following information is supplied to the computer by the data cards listed below:

<u>Card Nos.</u>	<u>Information</u>
3 & 4	Array of coefficients of plant differential equation (i.e., denominator of H(s)).
5 & 6	Array of coefficients of numerator of G(Z).

Card Nos.	Information
7 & 8	Array of values of zeros and poles of G(Z) to be included in overall process pulse transfer function, K(Z).
9 & 10	Array of coefficients of denominator of G(Z).
11 & 12	Array of coefficients of overall process pulse transfer function, K(Z).

(9) For certain investigations it may be desirable to limit the output of the digital controller to a signal of a given magnitude to avoid plant saturation. If so, the user must supply this value as a positive decimal value, E2MAX, and the program will limit the processed error from the controller to plus or minus this value. If no limiting is desired, a value of zero (0.0) should be given to E2MAX and the program then sets a magnitude limit of one million on the processed error. Data Card 13 contains the data on E2MAX.

NOTE 3: In very simple cases it may be that there are no zeros or poles of G(Z) that need to be included in K(Z). When this occurs, obviously the datum NZ is zero and the array ZERO(i) does not exist. Under these conditions, prepare data card 7 as shown below



and remove card 8 completely!

DATA CARDS for input to Program ALSTAPP

Card No.	Format	Contents	Remarks
1	2F10.0,I1	DELT, T, INPUT	Total time = 1000 DELT. Sample period T = 1/error sampling rate. DELT is the increment used for simulation of continuous plant. INPUT = 1 for a step, 2 for a ramp, 3 for acceleration.
2	I2,F18.0,I1	K, VK, NTYPE	K = order of plant. VK = plant gain. NTYPE = type of servo (i.e., 1,2...)
3	I2	NA	NA = no. terms in denominator of plant transfer function. Usually NA = K + 1. $H(S) = \dfrac{VK}{A(K+1)S^K + \ldots A(2)S+A(1)}$
4	8F10.0	A(I) I = 1, NA	A(I) is the array of coefficients of plant differential equation (i.e., denominator of H(S) above).
5	I2	NP	NP = no. terms in numerator of Z-transform of plant and zero-order hold whose z-transform is $G(Z) = \dfrac{P(Z)}{Q(Z)} =$
6	8F10.0	P(I) I=1, NP	$\dfrac{P(1)z^{-1}+P(2)z^{-2}+\ldots}{1+Q(2)z^{-1}+Q(3)z^{-2}+\ldots}$ $\ldots \dfrac{+P(NP)z^{-NP}}{+Q(NQ)z^{-NQ+1}}$
7	I2	NZ	NZ = no. zeros of G(Z) to be included in overall pulse transfer fcn, K(Z).

DATA CARDS for input to Program ALSTAPP (Continued)

Card No.	Format	Contents	Remarks
8	8F10.0	ZERO(I) I=1,NZ	ZERO(I) = array of values of such zeros of G(Z) as are to be included in K(Z).
9	I2	NQ	NQ = no. terms in denominator of G(Z).
10	8F10.0	Q(I) I=1, NQ	Q(I) = array of denominator coefficients.
11	I2	NK	NK = no. terms in overall pulse transfer function $K(Z)=AK(1)z^{-1}+AK(2)z^{-2} \ldots +AK(NK)z^{-NK}$
12	8F10.0	AK(I) I = 1, NK	AK(I) = array of process coefficients shown above.
13	F10.0	E2MAX	E2MAX is the limit placed on the absolute value of the output of the digital controller.

Notes to the User:

The FORMAT conveys information concerning the mode and location on the IBM card for each item of numerical information to be supplied to the program. If the following remarks are not sufficient, refer to the CDC Fortran Manual.

"I" means an integer value. Whole numbers only. Do not show a decimal point. The integer value must be "right justified" in its allotted field. If the field is left blank, a value of zero will be inferred.

"F" means a decimal value, either less than or greater than unity. A decimal point must be shown somewhere in the

allotted field. If the value is negative, the minus sign must also fall within the field. Precise location of the value (including sign, if any, and the decimal point <u>always</u>) within the field does not matter.

The numbers following I and F give the number of columns on an IBM card that are reserved for the field of that particular datum. For example, take the FORMAT control for data card number 2: Columns 1 thru 2 are reserved for the integer value of K. If the integer value of K is four, then card 2 should have 04 in its first two columns. The next eighteen columns (3 thru 20) on card number 2 must contain 1500000.0 somewhere between columns 3 and 20 inclusive. On both cards 1 and 2, column 21 must contain a single integer.

On those cards having an I2 format, columns 1 and 2 contain an integer value between one and ninety-nine. If the value is less than ten, it must appear in column 2 (i.e., the integer value must be <u>right</u> justified in its field. Decimal values need not be justified). In every case these I2 integer values tell the computer how many decimal values to read from the next card or cards. This brings us to the FORMAT 8F10.0. As before, the F10.0 means a decimal value with ten card columns reserved for it. 8F10.0 means there may be up to eight such values on a single card (a card has eighty columns), and the computer will read another card if it has to, in order to read as many values as were specified by the previous I2

card. Most input arrays will contain eight or less values, however, and thus will fit nicely on a single card. Note that these decimal fields of ten columns each begin in card columns 1, 11, 21, 31, ...etc. Card 1 has two decimal fields of ten columns each, followed by an integer field of one (1) in column 21. Card 13 has a single decimal field occupying card columns 1 thru 10.

If a decimal value of <u>zero</u> is to be entered, then place 0.0 in the appropriate field. For example, suppose array A has three decimal values A(1) = zero, A(2) = twenty, A(3) = minus two and a half. Then Card 4 should contain

Between Columns	The Decimal Value
1 and 10	0.0
11 and 20	20.0
21 and 30	-2.5

If no sign is shown, the value is taken as being positive.

In order to use this program to test a system using a <u>single-rate</u> controller, only the first thirteen data cards are required.

If you desire to test a system using a <u>double-rate</u> controller, then an additional four data cards are required. These cards contain information as to the double-rate controller constants and follow card 13 (<u>no</u> intervening blanks!) as shown below:

ADDITIONAL DATA CARDS FOR DOUBLE-RATE CONTROLLER

Card No.	Format	Contents	Remarks
14	I2	NAA	NAA is the no. terms in numerator of double-rate controller, $D(Z_2)$.
15	8F10.0	AA(I) I=1, NAA	Array of numerator coefficients.
16	I2	NBB	NBB is the no. terms in denominator of $D(Z_2)=\frac{AA(1)+AA(2)z^{-1}+\ldots+AA(NAA)z^{-NAA+1}}{BB(1)+BB(2)z^{-2}+\ldots BB(NBB)z^{-NBB+1}}$
17	8F10.0	BB(I) I=1, NBB	Array of denominator coefficients.

Example Problem using Program ALSTAPP:

$$H(S) = \frac{80,000}{S^3 + 95S^2 + 1750S}$$

By looking at H(S) we can see that

VK = 80000.0
K = 3
NA = 4
A(1) = 0.0
A(2) = 1750.0
A(3) = 95.0
A(4) = 1.0
NTYPE = 1

Let us choose to observe the response of this plant for one second after the application of a unit step input.

So DELT = 1/1000 = 0.001

INPUT = 1

Let us select a sampling rate of 100 times per second for the sampled data system, i.e., $T = 0.01$.

We prepare the first four data cards as shown below:

	Column 1	Column 11	Column 21	Column 31	
Card 4	0.0	1750.0	95.0	1.0	Card 4
Card 3	04				Card 3
Card 2	03	80000.0	1		Card 2
Card 1	0.001	0.01	1		Card 1

By partial fraction expansion and tables of Z-transforms we find the pulse transfer function of the plant-and-hold to be

<u>in factored form</u>:

$$G(Z) = \frac{.0106Z^{-1}\,(1 + 0.209327Z^{-1})\,(1 + 2.967275Z^{-1})}{(1-Z^{-1})\,(1-0.778801Z^{-1})\,(1 - 0.496585Z^{-1})}$$

<u>in polynomial form</u>:

$$G(Z) = \frac{.010600Z^{-1} + .033672Z^{-2} + .006584Z^{-3}}{1.0 - 2.275386Z^{-1} + 1.662127Z^{-2} - 0.386741Z^{-3}}$$

There is one zero outside the unit circle at -2.967275.

Choosing a minimal prototype process we find that

$$K(Z) = 0.252062Z^{-1} + 0.747938Z^{-2}$$

and we see the following data:

NP = 03	P(1) = .010600
	P(2) = .033672
	P(3) = .006584
NZ = 01	ZERO (1) = -2.967275

NQ = 04	Q(1) = 1.0
	Q(2) = -2.275386
	Q(3) = 1.662127
	Q(4) = -0.386741
NK = 02	AK(1) = 0.252062
	AK(2) = 0.747938

So we prepare data cards as follows:

	Column 1	Column 11	Column 21	Column 31
Card 12	0.252062	0.747938		
Card 11	02			
Card 10	1.0	-2.275386	1.662127	-0.386741
Card 9	04			
Card 8	-2.9667275			
Card 7	01			
Card 6	.010600	.033672	.006584	
Card 5	03			

Suppose we do not desire to limit the output of the digital controller. Then set E2MAX = zero by preparing card 13 as shown below:

	Column 1
Card 13	0.0

If it is not desired to simulate a double-rate digital controller, then only the first thirteen (13) data cards are required.

If double-rate action is desired, then the user must

supply the coefficients of the double-rate controller. The pulse transfer function of the double-rate controller has the following general form:

$$D(Z_2) = \frac{AA(1) + AA(2)Z_2^{-1} + AA(3)Z_2^{-2} + \ldots + AA(NAA)Z_2^{-NAA+1}}{BB(1) + BB(2)Z_2^{-1} + BB(3)Z_2^{-2} + \ldots + BB(NBB)Z_2^{-NBB+1}}$$

where all coefficients have been normalized with respect to BB(1) and BB(1) = 1.0.

The format for cards 14 thru 17 is identical to that for cards 5 thru 12.

In the double-rate controller the error-sampling period, T, remains the same as it was for the single-rate case. It is the output from the controller that occurs at the double-rate.

<u>Modification to allow for addition of white noise to the test input.</u>

When used as described in the main body of this appendix, the program simulates a control system whose input is completely deterministic. Gaussian noise of selected mean and variance may be added to the input by making an addition to the first data card and inserting a <u>new</u> data card between the previous card 1 and 2, as follows.

1. The presence of noise in the input must be requested by placing the <u>integer</u> 1 in column 31 of the first data card (which already contains DELT, T, and INPUT).

2. The new data card, which must follow card 1 above, contains an <u>integer</u> number between 01 and 99 right

justified in columns 1 and 2, and a decimal value in columns 3 through 20. The decimal value specifies the selected mean value of the noise, and in most cases will be shown as 0.0.

The integer number specifies the number of uniformly distributed random numbers from which each normally distributed random number is obtained. For a good approximation to white noise, this number should be between ten and twenty. The variance of the Gaussian distribution depends on this number, N, as shown below:

$$\text{Variance} = \frac{1}{N} \cdot \frac{1}{12}$$

As N approaches one, the noise approaches a uniform distribution between -1/2 and +1/2, for a selected mean of zero.

The remainder of the data cards are then included in exactly the order and format described earlier.

Modification to allow the program to use single-rate digital controller coefficients supplied by the user.

When used as described in the main body of this appendix, the program computes the coefficients of the single-rate controller using the following formula:

$$D(Z) = \frac{1}{G(Z)} \cdot \frac{K(Z)}{1 - K(Z)}$$

In order to do this, the user must supply data on the numerator and denominator of G(Z), the overall pulse transfer function, K(Z), and which zeros, if any, of G(Z) that are included in K(Z). CAUTION: Slide-rule accuracy

is usually not sufficient except in very simple cases! It is recommended that a desk calculator be used to provide this data accurately to at least five decimal places.

In case the user wishes to calculate his own controller coefficients for the single-rate sampled-data system simulation, he may cause the program to omit the calculation of D(Z) and to employ the controller supplied by the user. In order to do this, the data cards are arranged as shown below:

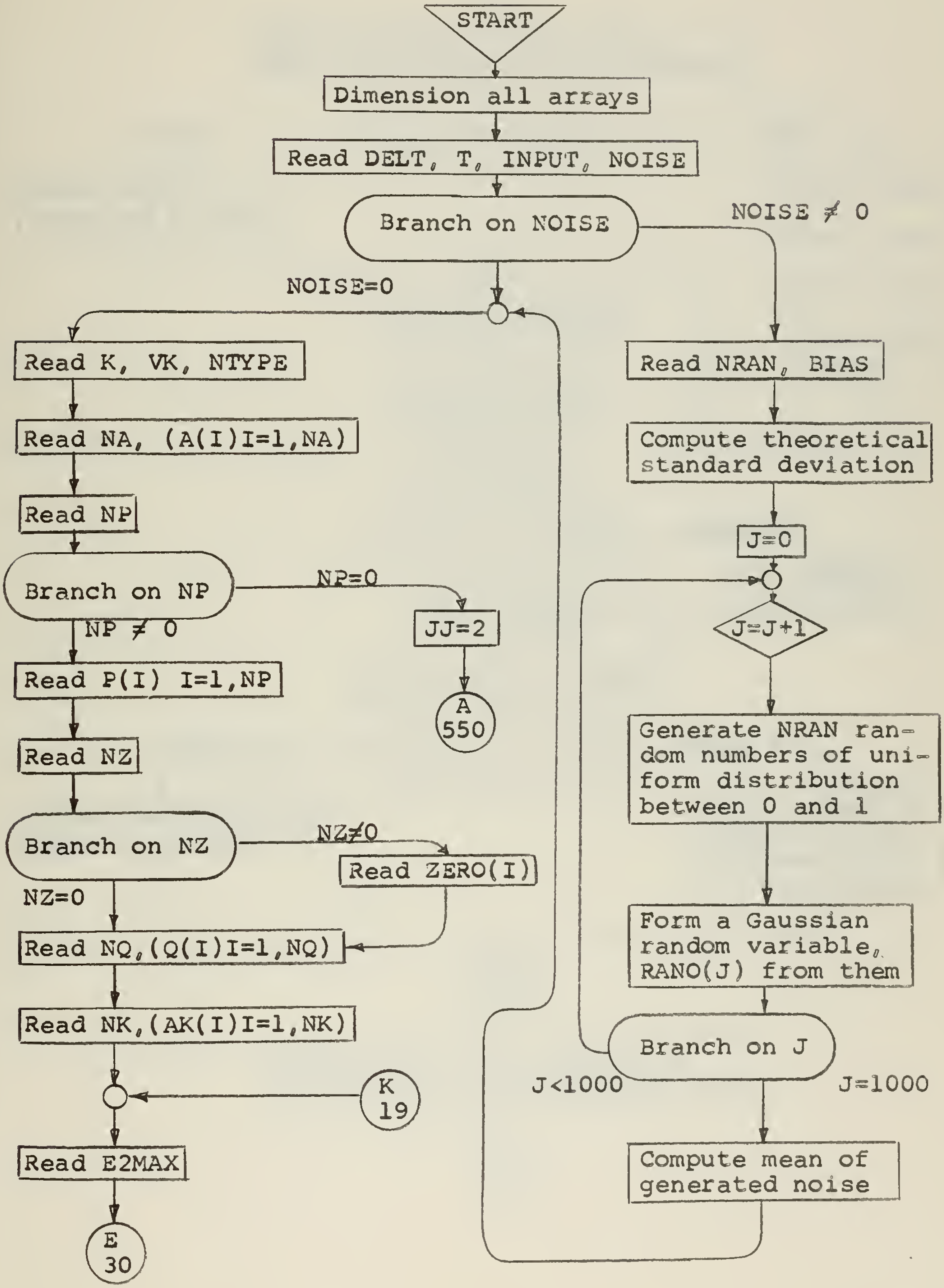
Card No.	Contents	Remarks
1	DELT, T, INPUT	Same as before
2	K, VK, NTYPE	Same as before
3	NA	Same as before
4	A(I) I = 1, NA	Same as before
5	Blank card!	Indicates D(Z) supplied by user
6	NAA	Single-rate controller coefficients
7	AA(I) I = 1, NAA	Single-rate controller coefficients
8	NBB	Single-rate controller coefficients
9	BB(I) I = 1, NBB	Single-rate controller coefficients
10	E2MAX	Same as before
11*	NAA	Double-rate controller coefficients
12*	AA(I) I = 1, NAA	Double-rate controller coefficients
13*	NBB	Double-rate controller coefficients

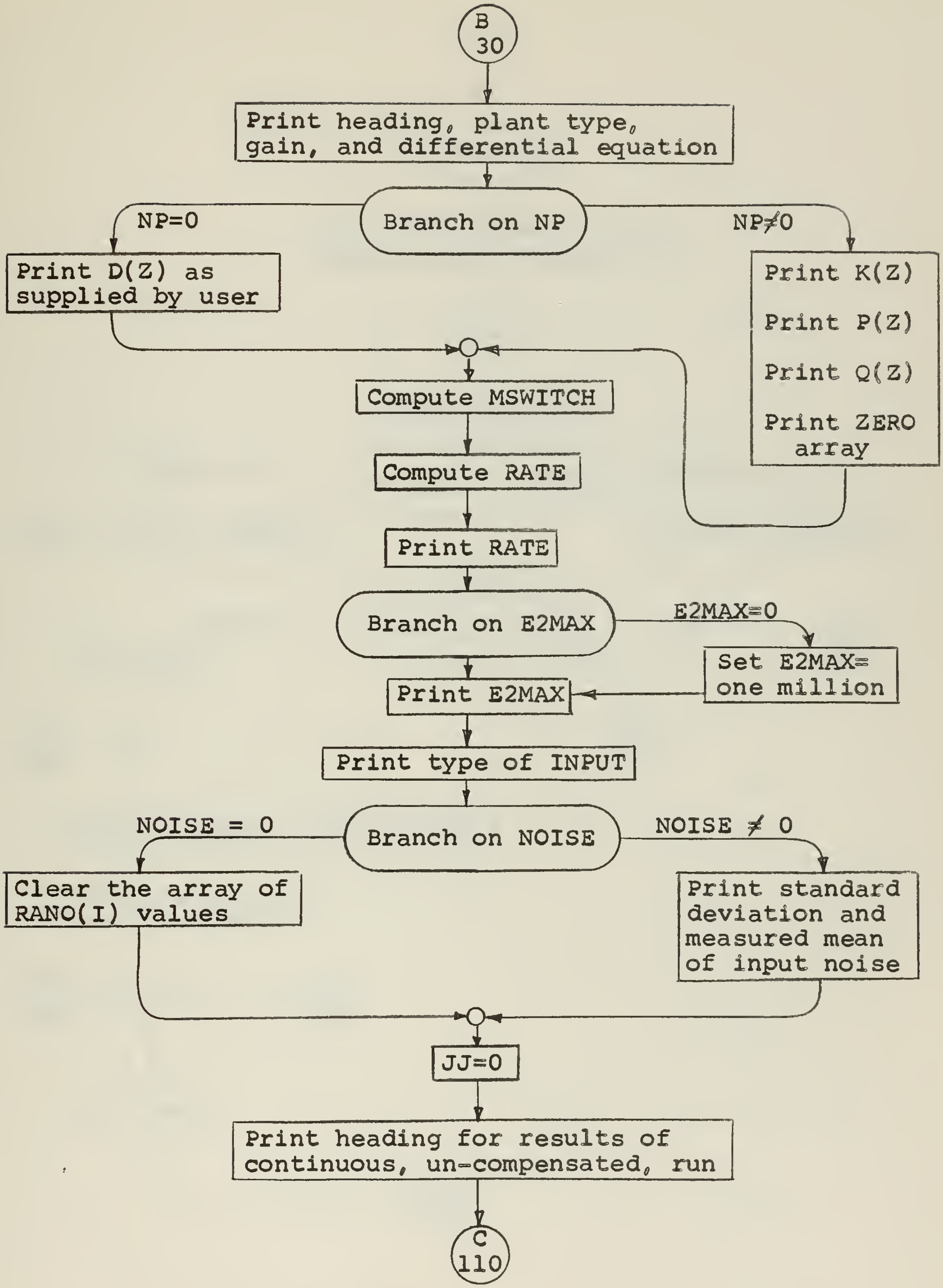
Card No.	Contents	Remarks
14*	BB(I) I = 1, NBB	Double-rate controller coefficients

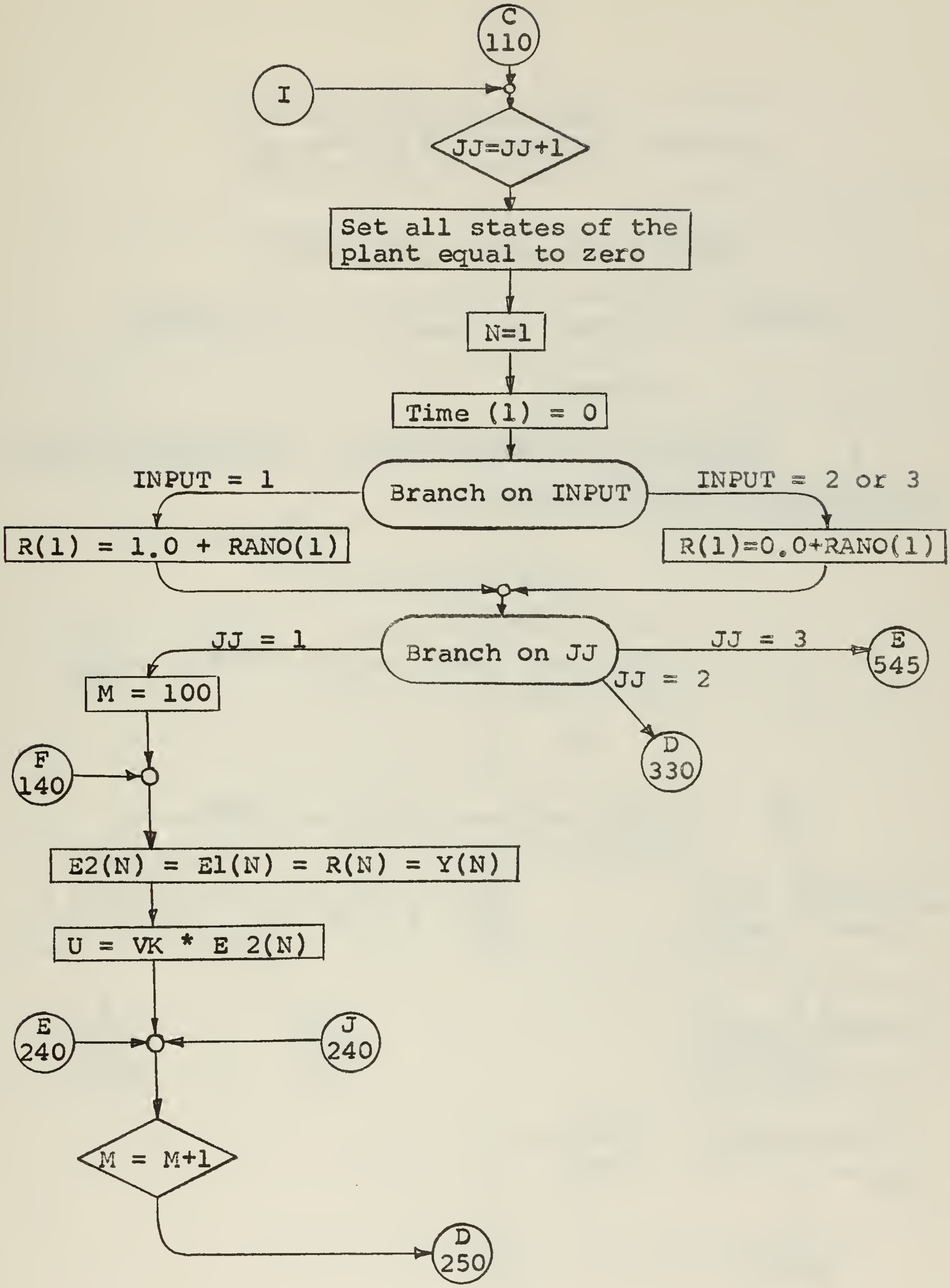
*Optional.

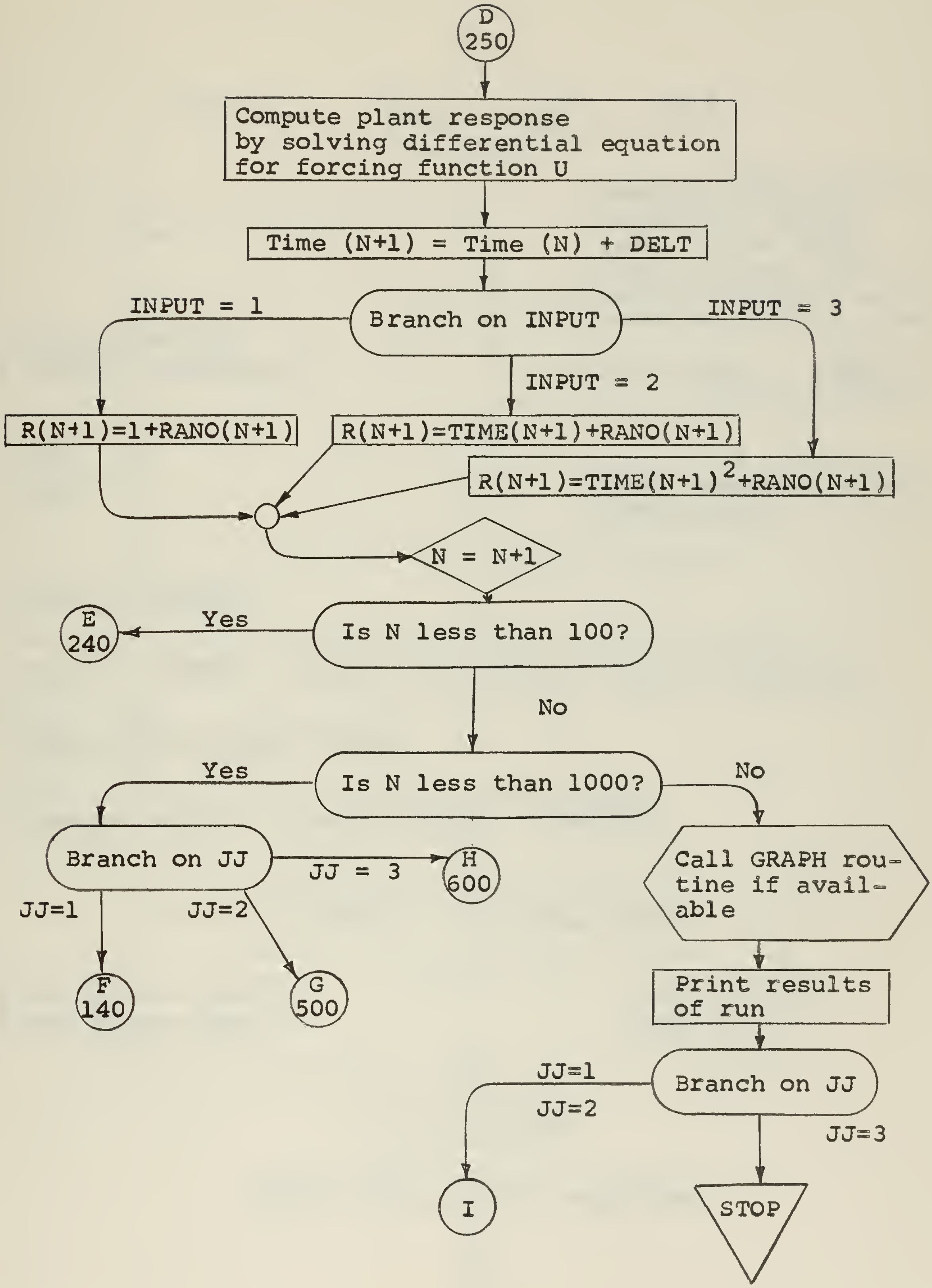
The first four cards are just the same as they would be for a deterministic input where D(Z) is to be computed by the program. A blank fifth card signals the program that D(Z) will be supplied by the next four cards. These next four cards (6, 7, 8, and 9) give the data for the single-rate controller exactly as described earlier in the main body of this memo for the double-rate controller coefficients. Card 10 gives E2MAX. If double-rate action is also desired, the last four cards must give the double-rate D(Z).

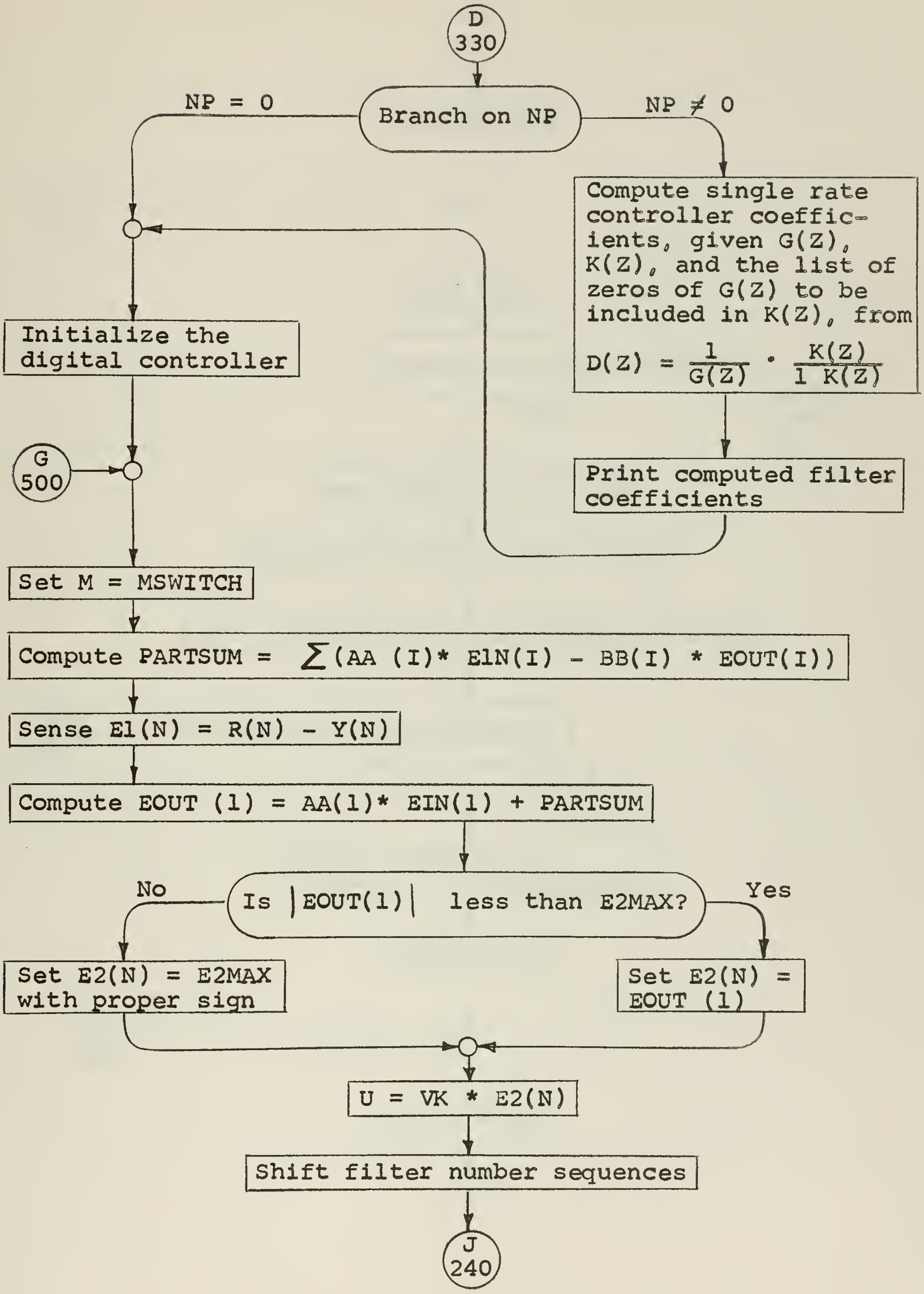
FLOW DIAGRAM for Program ALSTAPP

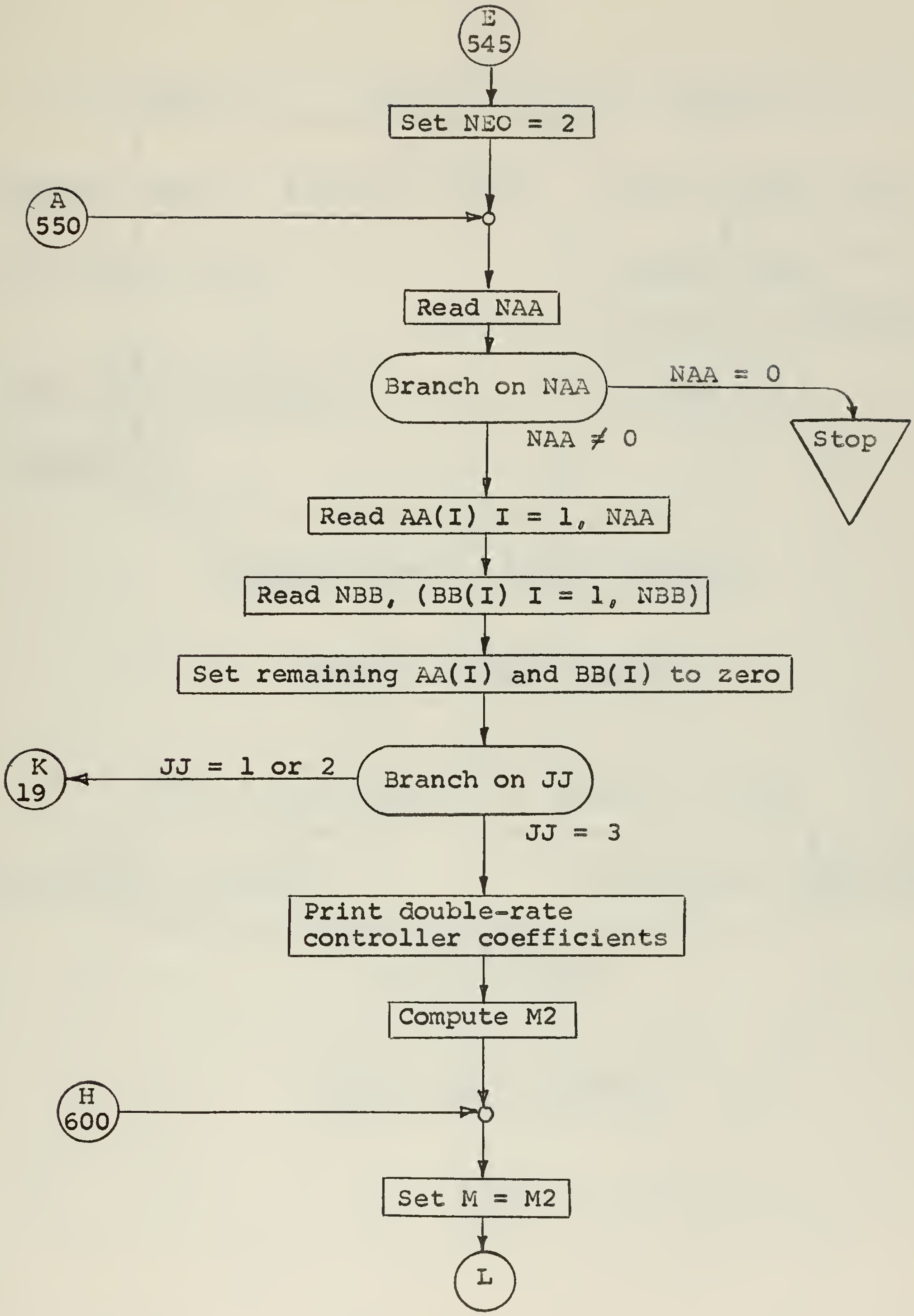


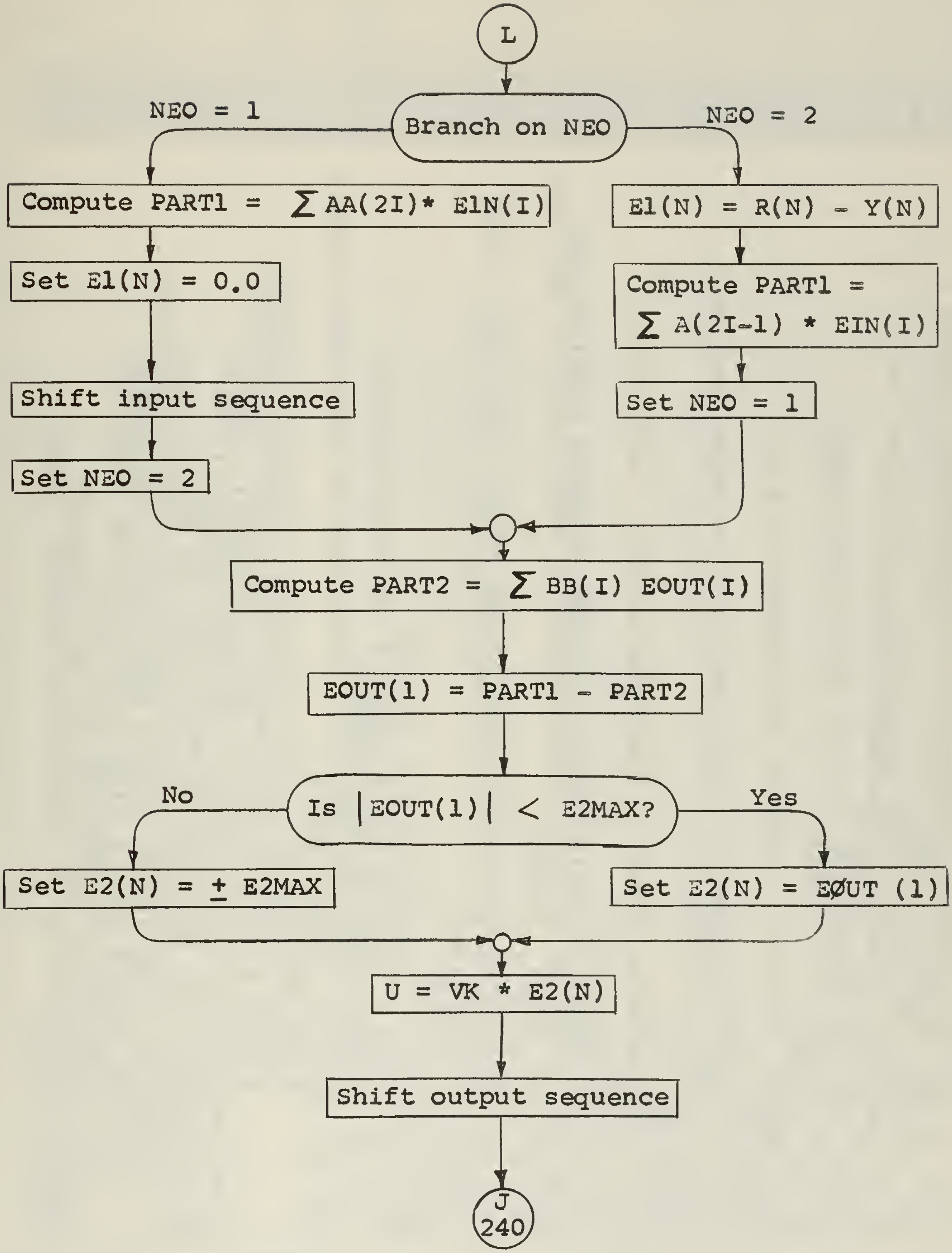

B
30
Print heading, plant type, gain, and differential equation
Branch on NP
NP=0
NP≠0
Print D(Z) as supplied by user
Print K(Z)
Print P(Z)
Print Q(Z)
Print ZERO array
Compute MSWITCH
Compute RATE
Print RATE
Branch on E2MAX
E2MAX=0
Set E2MAX= one million
Print E2MAX
Print type of INPUT
Branch on NOISE
NOISE = 0
NOISE ≠ 0
Clear the array of RANO(I) values
Print standard deviation and measured mean of input noise
JJ=0
Print heading for results of continuous, un-compensated, run
C
110


C
110
I
JJ=JJ+1
Set all states of the plant equal to zero
N=1
Time (1) = 0
INPUT = 1
Branch on INPUT
INPUT = 2 or 3
R(1) = 1.0 + RANO(1)
R(1)=0.0+RANO(1)
JJ = 1
Branch on JJ
JJ = 3
E
545
JJ = 2
D
330
M = 100
F
140
E2(N) = E1(N) = R(N) = Y(N)
U = VK * E 2(N)
E
240
J
240
M = M+1
D
250


D
250
Compute plant response by solving differential equation for forcing function U
Time (N+1) = Time (N) + DELT
Branch on INPUT
INPUT = 1
INPUT = 2
INPUT = 3
R(N+1)=1+RANO(N+1)
R(N+1)=TIME(N+1)+RANO(N+1)
R(N+1)=TIME(N+1)2+RANO(N+1)
N = N+1
Is N less than 100?
Yes
E
240
No
Is N less than 1000?
Yes
No
Branch on JJ
JJ = 3
H
600
JJ=1
JJ=2
F
140
G
500
Call GRAPH routine if available
Print results of run
Branch on JJ
JJ=1
JJ=2
JJ=3
I
STOP


D
330
Branch on NP
NP = 0
NP ≠ 0
Compute single rate controller coefficients, given G(Z), K(Z), and the list of zeros of G(Z) to be included in K(Z), from
$D(Z) = \frac{1}{G(Z)} \cdot \frac{K(Z)}{1 \; K(Z)}$
Print computed filter coefficients
Initialize the digital controller
G
500
Set M = MSWITCH
Compute PARTSUM = ∑(AA (I)* E1N(I) - BB(I) * EOUT(I))
Sense E1(N) = R(N) - Y(N)
Compute EOUT (1) = AA(1)* EIN(1) + PARTSUM
Is |EOUT(1)| less than E2MAX?
No
Yes
Set E2(N) = E2MAX with proper sign
Set E2(N) = EOUT (1)
U = VK * E2(N)
Shift filter number sequences
J
240


E 545
Set NEO = 2
A 550
Read NAA
Branch on NAA
NAA = 0
Stop
NAA ≠ 0
Read AA(I) I = 1, NAA
Read NBB, (BB(I) I = 1, NBB)
Set remaining AA(I) and BB(I) to zero
K 19
JJ = 1 or 2
Branch on JJ
JJ = 3
Print double-rate controller coefficients
Compute M2
H 600
Set M = M2
L


L
Branch on NEO
NEO = 1
NEO = 2
Compute PART1 = ∑ AA(2I)* E1N(I)
E1(N) = R(N) - Y(N)
Set E1(N) = 0.0
Compute PART1 = ∑ A(2I-1) * EIN(I)
Shift input sequence
Set NEO = 1
Set NEO = 2
Compute PART2 = ∑ BB(I) EOUT(I)
EOUT(1) = PART1 - PART2
Is |EOUT(1)| < E2MAX?
No
Yes
Set E2(N) = ± E2MAX
Set E2(N) = EØUT (1)
U = VK * E2(N)
Shift output sequence
J
240

```
      PROGRAM ALSTAPP
C     COMPUTER SIMULATION OF A SAMPLED-DATA CONTROL SYSTEM
C     PROGRAM SIMULATES THE OPERATION OF THE FOLLOWING
C     1.  THE CONTINUOUS, UN-COMPENSATED SYSTEM
C     2.  THE SAME SYSTEM WITH A SINGLE-RATE DIGITAL CONTROLLER
C     3.  OPTIONALLY, THE SYSTEM WITH A DOUBLE-RATE CONTROLLER
C     TEST INPUTS OF DETERMINISTIC UNIT STEP, RAMP, AND ACCELERATION
C     ARE AVAILABLE.  IF DESIRED, THE INPUT MAY BE CONTAMINATED
C     WITH A WHITE NOISE OF SELECTED MEAN AND VARIANCE.
      DIMENSION TIME(1300),R(1300),E1(1300),E2(1300),Y(1390),
     1          A(30), PY(30), Y1(30), Y2(30), DY1(30), DY2(30),
     2          P(15), Q(15), AK(20), BK(20), AD(20,40), BD(20,40),
     3          SYN(30), AA(20), BB(20), EIN(20), EOUT(20), ZERO(10),
     4          RANO(1001), X(30)
      READ 10, DELT, T, INPUT, NOISE
C     DELT IS THE TIME INCREMENT USED FOR SIMULATION OF CONTINUOUS PLANT
C     T IS THE SINGLE-RATE SAMPLING PERIOD
C     INPUT IS 1 FOR A STEP, 2 FOR A RAMP, 3 FOR ACCELERATION
C     NOISE = ZERO IF PURE INPUTS ARE DESIRED
   10 FORMAT(2F10.0, I1, I10)
      IF(NOISE) 21,11,21
   11 READ 12, K, VK, NTYPE
C     K IS THE ORDER OF PLANT, VK IS THE NUMERATOR GAIN CONSTANT
C     NTYPE IS THE NUMBER OF INTEGRATIONS IN THE PLANT ( IE. SERVO TYPE)
   12 FORMAT(I2, F18.0, I1)
      READ 13, NA, (A(I)  I=1,NA)
C     NA IS THE NUMBER OF TERMS IN DENOMINATOR OF PLANT TRANSFER FUNCTION
C     A(I) IS THE ARRAY OF COEFFICIENTS OF PLANT TRANSFER FUNCTION H(S)
   13 FORMAT(I2,/,(8F10.0))
      READ 14, NP
C     NP IS THE NUMBER OF TERMS IN NUMERATOR OF G(Z) = P(Z)/Q(Z)
   14 FORMAT(I2)
      IF(NP) 15,26,15
   15 READ 16, (P(I)  I=1,NP)
C     P(I) IS ARRAY OF NUMERATOR COEFFICIENTS IN ASCENDING INVERSE Z
   16 FORMAT((8F10.0))
      READ 14,NZ
C     NZ IS NUMBER OF PLANT ZEROES TO BE INCLUDED IN K(Z)
      IF(NZ) 17,18,17
   17 READ 16, (ZERO(I)  I=1,NZ)
C     ZERO(I) ARE THE VALUES (IF ANY) OF SUCH PLANT ZEROES
   18 READ 13, NQ, (Q(I)  I=1,NQ)
C     NQ IS NUMBER OF TERMS IN DENOMINATOR OF G(Z)
C     Q(I) IS ARRAY OF DENOMINATOR COEFFICIENTS (ASCENDING INVERSE Z)
      READ 13, NK, (AK(I)  I=1,NK)
C     NK IS NUMBER OF TERMS (DATA POINTS) IN CLOSED LOOP PROCESS, K(Z)
C     AK(I) IS ARRAY OF PROCESS COEFFICIENTS IN ASCENDING INVERSE Z
   19 READ 20, E2MAX
```

```
C     E2MAX IS THE UPPER LIMIT IMPOSED ON THE PROCESSED ERROR
   20 FORMAT(F10.0)
      GO TO 30
   21 READ 12, NRAN, BIAS, NOTHING
      X(1) = 0.533218 * T
      XNUM = NRAN
      K = NRAN - 1
      SIGMA = SQRTF(1.0/(12.0*XNUM))
      DO 24  J=1,1001
      DO 22  I=1,K
      X(I+1) = (X(I)+1.438218) * (X(I)+97.35214159)
      KX = X(I+1)
      XK = KX
   22 X(I+1) = X(I+1) - XK
      RANO(J) = 0.0
      DO 23  I=1,NRAN
   23 RANO(J) = RANO(J) + X(I)
      RANO(J) = RANO(J)/XNUM - 0.5 + BIAS
      X(1) = X(NRAN) + 0.1111111
   24 CONTINUE
      EXPECT = 0.0
      DO 25  I=1,1001
   25 EXPECT = EXPECT + RANO(I)/1001.0
      GO TO 11
   26 JJ = 2
      GO TO 550
   30 PRINT 31
   31 FORMAT(41H PROGRAM ALSTAPP-CONTROL SYSTEM SIMULATOR,//)
      PRINT 32, NTYPE, K, VK, (I, A(I),  I=1,NA)
   32 FORMAT(21H THE PLANT IS OF TYPE,I2,13H AND OF ORDER,I2,/,
     1        18H THE PLANT GAIN IS, F20.2,/,
     2        52H COEFFICIENTS OF THE PLANT DIFFERENTIAL EQUATION ARE,/,
     3       (10X,2HA(,I2,2H)=, F15.3))
      IF(NP) 33,40,33
   33 PRINT 34
   34 FORMAT(//,71H THE SINGLE-RATE CONTROLLER WILL BE COMPUTED BY THE P
     1ROGRAM, GIVEN THAT)
      PRINT 35, (I, AK(I),  I=1,NK)
   35 FORMAT(/,52H COEFFICIENTS OF OVERALL PULSE TRANSFER FUNCTION ARE,/
     1       ,(9X,3HAK(,I2,2H)=, F17.8))
      PRINT 36, (I, P(I),  I=1,NP)
   36 FORMAT(/,38H COEFFICIENTS OF NUMERATOR OF G(Z) ARE,/,
     1       (10X,2HP(,I2,2H)=, F17.8))
      PRINT 37, (I, Q(I),  I=1,NQ)
   37 FORMAT(/,40H COEFFICIENTS OF DENOMINATOR OF G(Z) ARE,/,
     1       (10X,2HQ(,I2,2H)=, F17.8))
      PRINT 38, (I,ZERO(I), I=1,NZ)
   38 FORMAT(/,36H ZEROES OF G(Z) INCLUDED IN K(Z) ARE,/,
     1       (7X,5HZERO(,I2,2H)=, F17.8))
```

```
      GO TO 43
   40 PRINT 41
   41 FORMAT(//,75H THE SINGLE-RATE CONTROLLER COEFFICIENTS WERE SUPPLIE
     1D BY THE USER, AND ARE,/)
      PRINT 42, (I, AA(I), I, BB(I),   I=1,20)
   42 FORMAT((4H AA(,I2,2H)=,F15.8,6X,4H BB(,I2,2H)=,F15.8),//)
   43 SWITCH = (100.0*DELT - T)/DELT
      IF(SWITCH) 44,45,45
   44 MSWITCH = SWITCH - 0.111111
      GO TO 46
   45 MSWITCH = SWITCH + 0.111111
   46 RATE = 1.0/T
      PRINT 47, RATE
   47 FORMAT(///,27H THE ERROR SAMPLING RATE IS,F8.2,17H TIMES PER SECON
     1D)
      IF(E2MAX) 49,48,49
   48 E2MAX = 1000000.0
   49 PRINT 50, E2MAX
   50 FORMAT(64H PROCESSED ERROR FROM THE CONTROLLER IS LIMITED TO PLUS
     1OR MINUS, F12.3,/)
      GO TO (51,53,55), INPUT
   51 PRINT 52
   52 FORMAT(30H THE TEST INPUT IS A UNIT STEP)
      GO TO 59
   53 PRINT 54
   54 FORMAT(30H THE TEST INPUT IS A UNIT RAMP)
      GO TO 59
   55 PRINT 56
   56 FORMAT(38H THE TEST INPUT IS A UNIT ACCELERATION)
   59 IF(NOISE) 62,60,62
   60 DO 61  I=1,1001
   61 RANO(I) = 0.0
      GO TO 108
   62 PRINT 63, SIGMA
   63 FORMAT(36H CONTAMINATED BY GAUSSIAN NOISE WITH,/,
     1        24H A STANDARD DEVIATION OF, F12.8)
      PRINT 64, EXPECT
   64 FORMAT(23H AND A MEASURED MEAN OF, F12.8,//)
  108 PRINT 109
  109 FORMAT(//,31X,45H RESPONSE OF CONTINUOUS, UN-COMPENSATED PLANT)
      JJ = 0
  110 JJ = JJ+1
C     INITIALIZE PLANT AND PROGRAM
      DO 120 I=1,K
      Y1(I) = 0.0
  120 DY1(I)= 0.0
      N = 1
      NFUSE = 1001
      DO 121  I=1,6500
```

```
 121  TIME(I) = 0.0
C     GENERATE TIME ARRAY
      DO 122  I=1,1001
 122  TIME(I+1) = TIME(I) + DELT
C     GENERATE ARRAY OF INPUT VALUES
      GO TO (123,125,127), INPUT
 123  R(1) = 1.0 + RANO(1)
      DO 124  I=1,1001
 124  R(I+1) = 1.0 + RANO(I+1)
      GO TO 129
 125  R(1) = 0.0 + RANO(1)
      DO 126  I=1,1001
 126  R(I+1) = TIME(I+1) + RANO(I+1)
      GO TO 129
 127  R(1) = 0.0 + RANO(1)
      DO 128  I=1,1001
 128  R(I+1) = TIME(I+1)**2 + RANO(I+1)
 129  GO TO (130,330,545), JJ
 130  M = 100
 140  E1(N) = R(N) - Y(N)
      E2(N) = E1(N)
      U = VK* E2(N)
 240  M = M+1
C     COMPUTE PLANT RESPONSE BY SOLUTION OF DIFFERENTIAL EQUATION
C     AFTER CHECKING FOR BLOWN FUSE DUE TO PLANT INSTABILITY
      IF(R(N)*5.0    - Y(N)) 245,249,243
 243  IF(R(N)*5.0    + Y(N)) 245,249,249
 245  NFUSE = N
      GO TO 290
 249  DO 250 I=1,K
 250  PY(I) = Y1(I) + DELT * DY1(I)
      PSUM = U - A(1) * PY(1)
      DO 251 I=2,K
 251  PSUM = PSUM - A(I) * PY(I)
      DO 252 I=1,K
      DY2(K+1-I) = 1.0/A(K+2-I) * PSUM
 252  PSUM = PSUM + A(K+1-I)*PY(K+1-I) - A(K+2-I)*DY2(K+1-I)
      DO 253 I=1,K
 253  Y2(I) = Y1(I) + DELT/2.0 * (DY1(I) + DY2(I))
      Y(N+1) = Y2(1)
      N = N + 1
      DO 260 I=1,K
      DY1(I) = DY2(I)
 260  Y1(I) = Y2(I)
      IF(M-100) 265,270,270
 265  E1(N) = 0.0
      E2(N) = 0.0
      GO TO 240
 270  CONTINUE
```

```
      IF(N-NFUSE) 280,280,290
  280 NFUSE = 1001
      GO TO (140,500,600),JJ
  290 PRINT 300,(TIME(N),R(N),E1(N),E2(N),Y(N),  N=1,NFUSE,10)
  300 FORMAT(/,15X,4HTIME,13X,5HINPUT,14X,9HRAW ERROR,5X,
     1        15HPROCESSED ERRJR, 13X,6HOUTPUT,/, (F20.4,4F20.6))
      IF(NFUSE-1001) 310,320,320
  310 PRINT 311
  311 FORMAT(//,119H THE PLANT RESPONSE IS HIGHLY UNSTABLE AND THE FUSE
     1BLEW WHEN THE MAGNITUDE OF THE OUTPUT EXCEEDED FIVE TIMES THE INPU
     2T)
C     SET OUTPUT EQUAL TO INPJT IN CASE OF BLOWN FUSE,(FOR GRAPH SAKE)
  315 DO 316  I=1,1001
  316 Y(I) = R(I)
  320 CONTINUE
C     CALL GRAPH OUTPUT AT THIS POINT IF DESIRED
      CALL GRAPH(900,TIME,Y,8)
      GO TO(110,110,999),JJ
  330 IF(NP) 331,490,331
C     BEGIN COMPUTATION OF COEFFICIENTS OF FILTER, D(Z)=AA(Z)/BB(Z)
C     FROM G(Z)=P(Z)/Q(Z) AND K(Z)=AK(Z)
  331 CONTINUE
      PRINT 332
  332 FORMAT(////,45H STEPS IN COMPUTATION OF D(Z) ARE SHOWN BELOW)
      DO 345 I = 1,20
      AA(I) = 0.0
  345 BB(I) = 0.0
C     PRIME NUMERATOR ARRAY WITH ZEROS
      KK = NK + NQ - 1
      DO 340 I=1,NQ
      DO 340 J=1,KK
  340 AD(I,J) = 0.0
      DO 350  I=1,NQ
      DO 350  J=1,NK
  350 AD(I,J-1+I) = Q(NQ+1-I) * AK(NK+1-J)
      DO 370 J=1,KK
      PSUM = 0.0
      DO 360 I=1,NQ
  360 PSUM = PSUM + AD(I,J)
  370 AD(NQ+1, J) = PSUM
      PRINT 371,(AD(NQ+1,KK+1-J),J=1,KK)
  371 FORMAT(  //,27H PRODUCT OF Q(Z) TIMES K(Z),/,(8F15.8))
      SYN(1) = AD(NQ+1, KK)
      KK = KK - 1
      DO 375  I=2,KK
  375 SYN(I) =AD(NQ+1, KK+2-I) + SYN(I-1)
      PRINT 377,(SYN(I),I=1,KK)
  377 FORMAT(//, 22H AFTER DIVIDING BY Z-1,/,(8F15.8))
      IF (INPUT-NTYPE) 381,382,382
```

```
381   INTEG = INPUT
      GO TO 384
382   INTEG = NTYPE
384   INP = INTEG
      GO TO (400,385,385), INP
385   KK = KK- 1
      DO 390  I = 2,KK
390   SYN(I) = SYN(I) + SYN(I-1)
      PRINT 397
397   FORMAT(//,33H ANOTHER DIVISION BY Z-1 WAS DONE)
      INP = INP + 1
      IF (4-INP) 400,385,400
400   CONTINUE
      IF(NZ) 405,412,405
405   CONTINUE
      DO 410  J = 1,NZ
      KK = KK - 1
      DO 410  I = 2,KK
410   SYN(I) = SYN(I) + SYN(I-1) * ZERO(J)
      PRINT 411,(SYN(I),I=1,KK)
411   FORMAT(  //,29H AFTER DIVIDING OUT LAST ZERO,/,(8F15.8))
412   CONTINUE
      DO 415  I=1,KK
415   AA(I) = SYN(I)
      NAA = KK
417   BK(1) = 1.0
      DO 420 I=1,NK
420   BK(I+1) = -AK(I)
      KB = NK + 1
      KK = NP + NK
      DO 425  I=1,NP
      DO 425  J = 1,KK
425   BD(I,J) = 0.0
      DO 430  I=1,NP
      DO 430  J=1,KB
430   BD(I,J-1+I) = P(NP+1-I) * BK(KB+1-J)
      DO 440 J=1,KK
      PSUM = 0.0
      DO 435 I=1,NP
435   PSUM = PSUM + BD(I,J)
440   BD(NP+1,J) = PSUM
      PRINT 444,(BD(NP+1,KK+1-J),J=1,KK)
444   FORMAT(////,29H PRODUCT OF P(Z) TIMES 1-K(Z),/,(8F15.8))
      SYN(1) = BD(NP+1,KK)
      KK = KK-1
      DO 445  I=2,KK
445   SYN(I) = BD(NP+1,KK+2-I) + SYN(I-1)
      PRINT 446,(SYN(I),I=1,KK)
446   FORMAT(  //,22H AFTER DIVIDING BY Z-1,/,(8F15.8))
```

```
      INP = INTEG
      GO TO (460,450,450), INP
450   KK = KK- 1
      DO 455 I=2,KK
455   SYN(I) = SYN(I) + SYN(I-1)
      PRINT 397
      INP = INP + 1
      IF(4-INP) 460,450,460
460   IF(NZ) 463,469,463
463   CONTINUE
      DO 465 J=1,NZ
      KK = KK - 1
      DO 465 I = 2,KK
465   SYN(I) = SYN(I) + SYN(I-1) * ZERO(J)
      PRINT 466,(SYN(I),I=1,KK)
466   FORMAT(  //,29H AFTER DIVIDING OUT LAST ZERO,/,(8F15.8))
469   DO 470  I=1,KK
470   BB(I) = SYN(I)
      NBB = KK
475   CONTINUE
      BB1 = BB(1)
      DO 480  I=1,20
      AA(I) = AA(I) / BB1
480   BB(I) = BB(I) / BB1
      PRINT 484
      PRINT 485, ( I, AA(I), I, BB(I),    I=1,20)
484   FORMAT(/////,35H THE PROGRAM COMPUTED THE FOLLOWING)
485   FORMAT(    36H SINGLE RATE CONTROLLER COEFFICIENTS,/,(4H AA(,I2,
     1  2H)=,F12.8,6X,4H BB(,I2,2H)=,F12.8))
490   PRINT 494, T
494   FORMAT(////,31X,47H RESPONSE USING SINGLE-RATE CONTROLLER WITH T =
     1          ,F6.3,8H SECONDS)
495   CONTINUE
C     INITIALIZE FILTER INPUT AND OUTPUT NUMBER SEQUENCES
      DO 497 I=1,20
      EIN(I) = 0.0
497   EOUT(I)= 0.0
500   M = MSWITCH
C     SIMULATE DIGITAL COMPENSATION BY SINGLE RATE CONTROLLER
      PARTSUM = 0.0
      DO 510  I=2,NAA
510   PARTSUM = PARTSUM + AA(I)*EIN(I)
      DO 520  I=2,NBB
520   PARTSUM = PARTSUM - BB(I)*EOUT(I)
      EIN(1) = R(N) - Y(N)
      E1(N) = EIN(1)
      EOUT(1) = AA(1)*EIN(1) + PARTSUM
      IF(E2MAX - EOUT(1)) 521,521,523
521   E2(N) = +E2MAX
```

```
      GO TO 528
  523 IF(E2MAX + EOUT(1)) 525,525,527
  525 E2(N) = -E2MAX
      GO TO 528
  527 E2(N) = EOUT(1)
  528 U = VK*E2(N)
C     SHIFTING NUMBER SEQUENCES
      DO 530  I=1,19
  530 EIN(21-I) = EIN(20-I)
      DO 540 I=1,19
  540 EOUT(21-I) = EOUT(20-I)
      GO TO 240
  545 NEO = 2
C     SET FILTER COEFFICIENTS (AA(I),BB(I)) FOR DOUBLE-RATE CONTROLLER
  550 CONTINUE
      READ 14, NAA
      IF(NAA) 999,999,552
  552 READ 16, (AA(I)  I=1,NAA)
      KAA=NAA+1
      DO 560 I=KAA,20
  560 AA(I)=0.0
      READ 13, NBB, (BB(I)  I=1,NBB)
      KBB=NBB+1
      DO 570 I=KBB,20
  570 BB(I)=0.0
      GO TO (19,19,574), JJ
  574 PRINT 575, (I, AA(I), I, BB(I),  I=1,20)
  575 FORMAT(/////,8X,43H USER-SUPPLIED DOUBLE-RATE COEFFICIENTS ARE,/,
     1       (9X,3HAA(,I2,2H)=,F15.8,6X,3HBB(,I2,2H)=,F15.8))
C     INITIALIZE FILTER INPUT AND OUTPUT NUMBER SEQUENCES
      DO 580 I=1,20
      EIN(I) = 0.0
  580 EOUT(I) = 0.0
      HALFT = T/2.0
      PRINT 581, HALFT
  581 FORMAT(////,31X,49H RESPONSE USING DOUBLE-RATE CONTROLLER WITH T/2
     1 =, F6.4,8H SECONDS)
      M2 = 50.111111 + SWITCH/2.0
  600 M = M2
      GO TO (625,615), NEO
C     SIMULATE DIGITAL COMPENSATION BY DOUBLE-RATE CONTROLLER
  615 EIN(1) = R(N) - Y(N)
      E1(N) = EIN(1)
      PART1 = 0.0
      DO 620 I=1,10
  620 PART1 = PART1 + AA(2*I-1) * EIN(I)
      NEO = 1
      GO TO 645
  625 PART1 = 0.0
```

```
      DO 630 I=1,10
  630 PART1 = PART1 + AA(2*I) * EIN(I)
      E1(N) = 0.0
C     SHIFT OLD INPUT SEQUENCE IN PREPARATION FOR OTHER BRANCH
      DO 640 I=1,9
  640 EIN(11-I) = EIN(10-I)
      NEO = 2
  645 CONTINUE
      PART2 = 0.0
      DO 650  I=2,NBB
  650 PART2 = PART2 + BB(I) * EOUT(I)
      EOUT(1) = PART1 - PART2
      IF(E2MAX - EOUT(1)) 651,651,653
  651 E2(N) = +E2MAX
      GO TO 658
  653 IF(E2MAX + EOUT(1)) 655,655,657
  655 E2(N) = -E2MAX
      GO TO 658
  657 E2(N) = EOUT(1)
  658 U = VK*E2(N)
C     SHIFT OUTPUT NUMBER SEQUENCE EACH TIME
      DO 660 I=1,19
  660 EOUT(21-I) = EOUT(20-I)
      GO TO 240
  999 STOP
      END
      END
```

APPENDIX II

DIGITAL COMPUTER PROGRAM FOR FINDING Z-TRANSFORMS AND ROOTS OF POLYNOMIALS

General Description of Program Stappl

This program is a combination rootfinder and Z-transform taker. It will operate in one of three modes, as specified by the user. These modes are:

Mode 1

Program operates as a rootfinder only. The user supplies the degree and coefficients of the polynomial whose roots are to be found. Handles polynomials up to and including thirtieth degree.

Mode 2

It is desired to take the Z-transform of a transfer function which is not known in factored form, such as

$$H(s) = \frac{VK}{A(N+1)s^{N}+A(N)s^{N-1}+ \ldots + A(2)s + A(1)}$$

The program first finds the roots of the denominator polynomial. Then it takes the Z-transform of the above plant transfer function preceded by a zero-order hold, i.e.

$$G(Z) = \mathcal{Z}\left(\frac{1-e^{-st}}{s}.H(s)\right) = (1-Z^{-1}).\mathcal{Z}\,\frac{H(s)}{s} = \frac{P(Z)}{Q(Z)}$$

and presents the z-plane poles of G(Z) as well as the numerator and denominator polynomials in Z^{-1}. Then it operates as a rootfinder on the numerator, P(Z), in order to display the zeros of G(Z).

Mode 3

It is desired to find the Z-transform of a plant and

[illegible]

hold combination when the transfer function of the plant is known in factored form, such as

$$H(s) = \frac{VK}{s(s-P_1)(s-P_2)\ (s-P_3)\ \ldots\ (S-P_n)}$$

As in Mode 2, the Z-transform is taken with the plant preceded by a zero-order hold, i.e.

$$G(Z) = (1-Z^{-1}) \cdot \mathcal{Z}\ \frac{H(s)}{s} = \frac{P(Z)}{Q(Z)}$$

The program displays the Z-plane poles of G(Z) and the numerator and denominator polynomials. It then operates as a rootfinder on the numerator polynomial, P(Z), in order to display the zeros of G(Z).

<u>Restrictions</u>.

Mode 1 is relatively unrestricted. It will find the roots (real and imaginary parts) of polynomials in the following form (for N up to and including 30):

$$P(x) = A(N+1)\ x^N + A(N)x^{N-1} + \ldots +A(2)x + A(1)$$

Modes 2 and 3 are restricted to polynomials of degree ten or less, with all roots lying on the real axis and no repeated roots. Mode 2 will accept a polynomial whose eleventh root is at the origin. In both modes the Z-transform operation is predicated upon the plant being a type 1 servo (i.e. the Laplace transfer function has a single pole at the origin).

Since the program will not take the Z-transform when roots are complex, Mode 2 checks the magnitude of each imaginary part with respect to the corresponding real part of each root it finds. If the magnitude of any

imaginary part exceeds 1/100th of the corresponding real part, the program stops. If all imaginary parts of the roots found in the first phase of Mode 2 are less than 1/100th the magnitude of their corresponding real parts, then they are considered negligible and the root is treated as entirely real. The program then proceeds as described earlier.

If, thru error, Mode 2 is selected when the unfactored denominator of H(s) is higher than degree ten (exclusive of root at origin) then the program stops after finding the roots of the denominator.

Mathematical Methods: Root-finder

The rootfinding techniques is a modification of Bairstow's iterative procedure for finding a quadratic divisor of the form x^2-PP x-QQ. When this divisor is found sufficiently closely, it is solved by the quadratic formula, and divided out to leave a reduced polynomial two degrees lower. The criterion of suitable closeness is applied first to the change, DPP, in PP and then, if PP is sufficiently accurate, to the change, DQQ, in QQ. In both cases it is considered sufficiently accurate when the magnitude of the next correction to the coefficient is less than 10^{-10} of the magnitude of the coefficient itself.

The essence of Bairstow's iterative procedure is to divide a polynomial, say A(x), by the trial quadratic divisor to obtain a first quotient polynomial, say Q(x), plus a first remainder. Then this first quotient is

divided again by the same trial divisor to obtain a second quotient, say T(x), plus a second remainder. Then the corrections necessary in the coefficients of the trial quadratic divisor are determined from a suitable combination of the coefficients of the first and second remainders. A symbolic example will illustrate the method.

Let $F(x) = A(N+1)x^{N}+A(N)x^{N-1}+ \ldots +A(2)x+A(1)$ = the polynomial whose roots are to be found.

Let the trial quadratic divisor be $TD(x) = x^{2}-PP.x-QQ$

Now $F(x) = TD(x)\cdot\left[Q(N+1x^{N-1}+Q(N)x^{N-2}+\ldots+Q(4)x+Q(3)\right]$
$+Q(2)x+Q(1)$

where the first quotient is Q(x) inside the brackets and the first remainder is Q(2)x+Q(1).

Dividing again by the trial quadratic divisor:

$Q(x) = TD(x)\cdot\left[T(N-1)x^{N-3}+T(N-2)x^{N-4}+\ldots+T(4)x+T(3)\right]$
$+T(2)x+T(1)$

where the second remainder is seen to be T(2)x+T(1)

The corrections to the coefficients of TD(Z) are computed from the following formulae:

$$M = PP.T(1) + QQ.T(2)$$

$$D = T(1)^{2}-M.T(2)$$

$$DPP = \frac{T(2)\cdot Q(1) - T(1)\cdot Q(2)}{D}$$

$$DQQ = \frac{M.Q(2) - T(1)\cdot Q(1)}{D}$$

The modification of Bairstow's procedure came about as the result of several runs using test polynomials. It was found that convergence in the case of non-repeated roots was more consistently obtained if each set of iterations on successive reduced polynomials was begun from the origin. In case there are repeated roots the return

to the origin is delayed until all roots at that location have been found. It was also found that an occasional set of iterations would not terminate even though no further refinements were being made in the coefficients of the trial divisor. When this occurs the program assumes the trial divisor is satisfactory after fifty iterations. No degradation of accuracy has been observed due to this artificial convergence device.

Because of the sharpness of the criterion for a satisfactory trial divisor it appears that accuracy of root values to at least seven significant figures can be expected. Since the sum of the roots of an Nth degree polynomial is the coefficient of the term containing the variable to the (N-1)th power, the program calculates a check sum from the root values found and displays this check sum along with the true value as an indication of the accuracy achieved. This also provides the user with the means to tell at a glance whether or not the procedure converged.

Mathematical Methods: Z-transformer

The Z-transform portion of the program is restricted to type 1 servos preceded by a zero order hold. That is, the general form of the function whose Z-transform is to be taken must be

$$G(s) = \frac{1-e^{-sT}}{s} \cdot \frac{VK}{s(s-P_1)(s-P_2)\ldots(s-P_N)}$$

Where s is the Laplace variable, T is the sampling period,

and N is the number of poles on the real axis, <u>not</u> including the pole at the origin.

<u>Note</u>:

All poles must be real and none can be repeated.

By definition, $Z=e^{sT}$, so by partial fraction expansion we can write

$$G(Z) = VK\cdot\frac{(Z-1)}{Z}\mathcal{Z}\left(\frac{A}{s^2} + \frac{B}{s} + \frac{C(1)}{s-P_1} + \frac{C(2)}{s-P_2} + \cdots \frac{C(N)}{s-Pn}\right)$$

The program first computes the partial fraction coefficients, A, B, and C(i) i = 1, 2, ..., N. The Z-transform of each term is then taken as follows:

$$\mathcal{Z}\left(\frac{A}{s^2}\right) = \frac{ATZ}{(Z-1)^2}$$

$$\mathcal{Z}\left(\frac{B}{s}\right) = \frac{B}{Z-1}$$

$$\mathcal{Z}\left(\frac{C(i)}{S-Pi}\right) = \frac{C\ (i)}{Z-e^{PiT}}$$

The poles of G(Z), exclusive of the one at unity, are seen to be at $Z = e^{PiT}$ for i = 1, 2, ..., N

The program then proceeds to combine the individual Z-transforms over a common denominator, cancel the external factor $(1-Z^{-1})$, and multiply through by the plant gain constant, VK.

The final results are displayed as the coefficients of the numerator and denominator polynomials shown below:

$$G(Z) = \frac{P(1)Z^N + P(2)Z^{N-1} + P(3)Z^{N-2} + \ldots + P(N)Z + P(N+1)}{Q(1)Z^{N+1} + Q(2)Z^N + Q(3)Z^{N-1} + \ldots + Q(N+1)Z + Q(N+2)}$$

or, equivalently:

$$G(Z) = \frac{P(1)Z^{-1} + P(2)Z^{-2} + \ldots + P(N)Z^{-N} + P(N+1)Z^{-(N+1)}}{Q(1)+Q(2)Z^{-1}+Q(3)Z^{-2}+ \ldots +Q(N+1)Z^{-N}+Q(N+2)Z^{-(N+1)}}$$

Discussion of the Data Provided by the User

The general nature of the input data for each mode is discussed below. The sequence in which the data are presented is vital, and is correctly shown in the discussion.

Mode 1:

1) MODE = the integer number 1.

2) N = an integer value between 1 and 30 representing the degree of the polynomial whose roots are to be found.

3) A(i) = a linear array of decimal values representing the coefficients of the polynomial whose roots are to be found. These values must appear on the data card(s) from left to right in the same order as they are in the polynomial when written in descending powers of the variable. Example for N = 4:

$$A(5)x^4 + A(4)x^3 + A(3)x^2 + A(2)x + A(1) = F(x)$$

Mode 2:

1) MODE = the integer number 2.

2) N = an integer value between 1 and 10 representing the degree of the denominator polynomial of the plant transfer function, H(s).

3) A(i) = a linear array of decimal values representing the coefficients of the denominator of H(s) in descending powers of A. Example for N = 5:

$$H(s) = \frac{VK}{A(6)s^5 + A(5)s^4 + A(4)s^3 + A(3)s^2 + A(2)s + A(1)}$$

For a type 1 servo, A(1) = 0.0 and N is allowed to be as high as 11. The pole at the origin will be removed by the program and N will be reduced by 1 when a zero value is read for A(1). It is optional with the user whether or not he removes the pole at the origin himself, or allows the program to do it.

4) PERIOD and VK = both are <u>decimal</u> values. PERIOD is the reciprocal of the sampling rate for which the Z-transfer is to be taken. VK is the numerator gain constant of the plant transfer function, H(s).

<u>Mode 3</u>:

1) MODE = the <u>integer</u> number 3.

2) N = the <u>integer</u> number of poles of the plant transfer function, exclusive of the pole at the origin.

3) P(i) = a linear array of the <u>decimal</u> values of the poles of H(s). Does not include the pole at the origin. All poles must be simple and must lie on the real axis of the s-plane.

4) PERIOD and VK = same as in Mode 2 discussed above.

<u>Detailed Instructions for Preparing Input Data</u>

All input data are provided to the program through the use of IBM cards. The "normal" number of data cards required by each mode of operation is tabulated below:

Mode	No. of data cards required
1	3
2	4
3	4

"Normally" it is expected that the number of decimal values in the linear array will be eight (8) or less, so that the entire array can be contained on a single card. For arrays of more than eight values, additional cards are used as necessary.

For convenience, the following table summarizes the requirements for input data:

TABLE A

Card No.	Datum	Format	Used in		
			Mode 1	Mode 2	Mode 3
1	MODE	I1	X	X	X
2	N	I2	X	X	X
3*	(Array)	8F10.0	X	X	X
4	PERIOD, VK	F10.0, F20.0		X	X

* Use additional cards 3 as necessary for arrays containing more than eight (8) values.

The "Format" information needs to be explained. The format letter I means that the value is an integer. Integers must be shown without any decimal point and must be "right justified" in their allotted space. Right justified means that the right-most digit of the integer value must appear in the right-most column of the space allotted for that value on the card. This applies only to integer values.

The format letter F indicates a decimal value and a decimal point <u>must</u> be <u>used</u>. If the value is negative, the minus sign must appear. Unsigned values are taken

to be positive. The decimal value, including the decimal point always and minus sign if necessary, may appear anywhere within the allotted space (or "field").

The number following the letter I or F is the number of columns on the IBM card forming the allotted space or field for that value. There are eighty (80) columns on a card and they are assigned from left to right in order.

When a number <u>precedes</u> the format letter, it tells how many times that particular field is repeated. Thus card 3 contains eight identical fields of ten columns each. The next card may be read by the program as a continuation of card 3 if more than eight values are in the array.

Card 1 uses only the first column. This column must contain either the digit 1, 2, or 3.

Card 2 uses only the first and second columns. If the value of N is nine or less, then skip column one and place the digit in column two.

Card 3 uses as many identical fields of ten columns each as are necessary. Note that these fields begin in columns 1, 11, 21, 31, etc.

Card 4 uses the first ten columns to contain the decimal value for the PERIOD. The next twenty columns (i.e., 11 through 30) are reserved for the decimal value of VK. Decimal values may lie anywhere within their allotted field as long as the decimal point is shown in the field and the minus sign is in the field for negative values.

Let us now show the preparation of data cards for some representative problems.

<u>EXAMPLE PROBLEM</u> - <u>Mode 1</u>

Find all the roots of

$$F(x) = x^7 + 16x^6 + 102x^5 + 371x^4 + 838x^3 + 1200x^2 + 1012x + 420$$

Since all we need is a rootfinding operation, we choose Mode 1.

The degree, N, of the polynomial is 7. The polynomial is written in descending powers of x so the three data cards would be

Column 1	Column 11	Column 21	Column 31	Column 41	Column 51	Column 61	Column 71	
1.0	16.0	102.0	371.0	838.0	1200.0	1012.0	420.0	Card 3
07								Card 2
1								Card 1

The answers are:

Roots 1 and 2 = $-1.0 \pm j\,1.0$

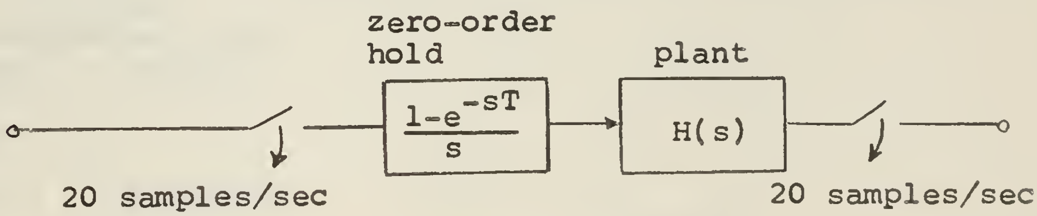
3 and 4 = $-1.5 \pm j\,1.658312$

5 and 6 = $-2.0 \pm j\,1.414214$

7 = -7.0

<u>EXAMPLE PROBLEM</u> - <u>Mode 2</u>

Find the Z-transform of the following configuration



where the plant transfer function is

$$H(s) = \frac{1.6064235 \times 10^9}{s^5 + 134s^4 + 5064s^3 + 73274s^2 + 356983s}$$

Since H(s) is not known in factored form we choose Mode 2.

The degree, N, of the denominator of H(s) is 5*.

The denominator of H(s) is written in descending powers. Note that A(1) is missing!

The numerator gain constant, VK, is 1.6064235×10^9.

The sampling rate is twenty samples per second so the sampling PERIOD is .05 seconds.

The data cards are prepared as shown below:

Column 1	Column 11	Column 21	Column 31	Column 41	Column 51	Column 61	Column 71	
.05	1606423500.0							Card 4
1.0	134.0	5064.0	73274.0	356983.0	0.0			Card 3
05								Card 2
2								Card 1

* User may remove pole at origin, if desired, and enter a 4th degree denominator where A(1) = 356983.

The answers are:

$$\text{Z-transform} = G(Z) = \frac{P(Z)}{Q(Z)} = \frac{P(1)Z^{-1}+P(2)Z^{-2}+P(3)Z^{-3}+P(4)Z^{-4}+}{Q(1)+Q(2)Z^{-1}+Q(3)Z^{-2}+Q(4)Z^{-3}+}$$

$$\frac{+\ P(5)Z^{-5}}{Q(5)Z^{-4} + Q(6)Z^{-5}}$$

where

P(1) = 1.54749758 Q(1) = 1.0

P(2) = 16.74877866 Q(2) = -2.33676594

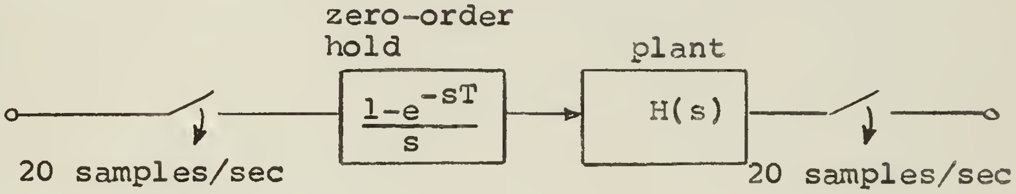
P(3) = 16.29465286	Q(3) = 1.92220653
P(4) = 2.04809669	Q(4) = - .67242310
P(5) = .01866472	Q(5) = .08821342
	Q(6) = - .00123091

The zeros of G(Z) are at Z = - .00988230
- .13606251
- .91925736
-9.75793506

The poles of G(Z) are at Z = 0.57694981
0.42741498
0.31663673
0.01576442

EXAMPLE PROBLEM - Mode 3

Find the Z-transform of the following configuration:



where the plant transfer function is

$$H(s) = \frac{1.6064235 \times 10^9}{s(s + 11)(s + 17)(s + 23)(s + 83)}$$

Since H(s) is known in factored form we choose Mode 3.

The number, N, of real poles (not counting the one at the origin) is 4.

The numerator gain constant, VK, is 1.6064235×10^9.

The sample PERIOD is .05 seconds.

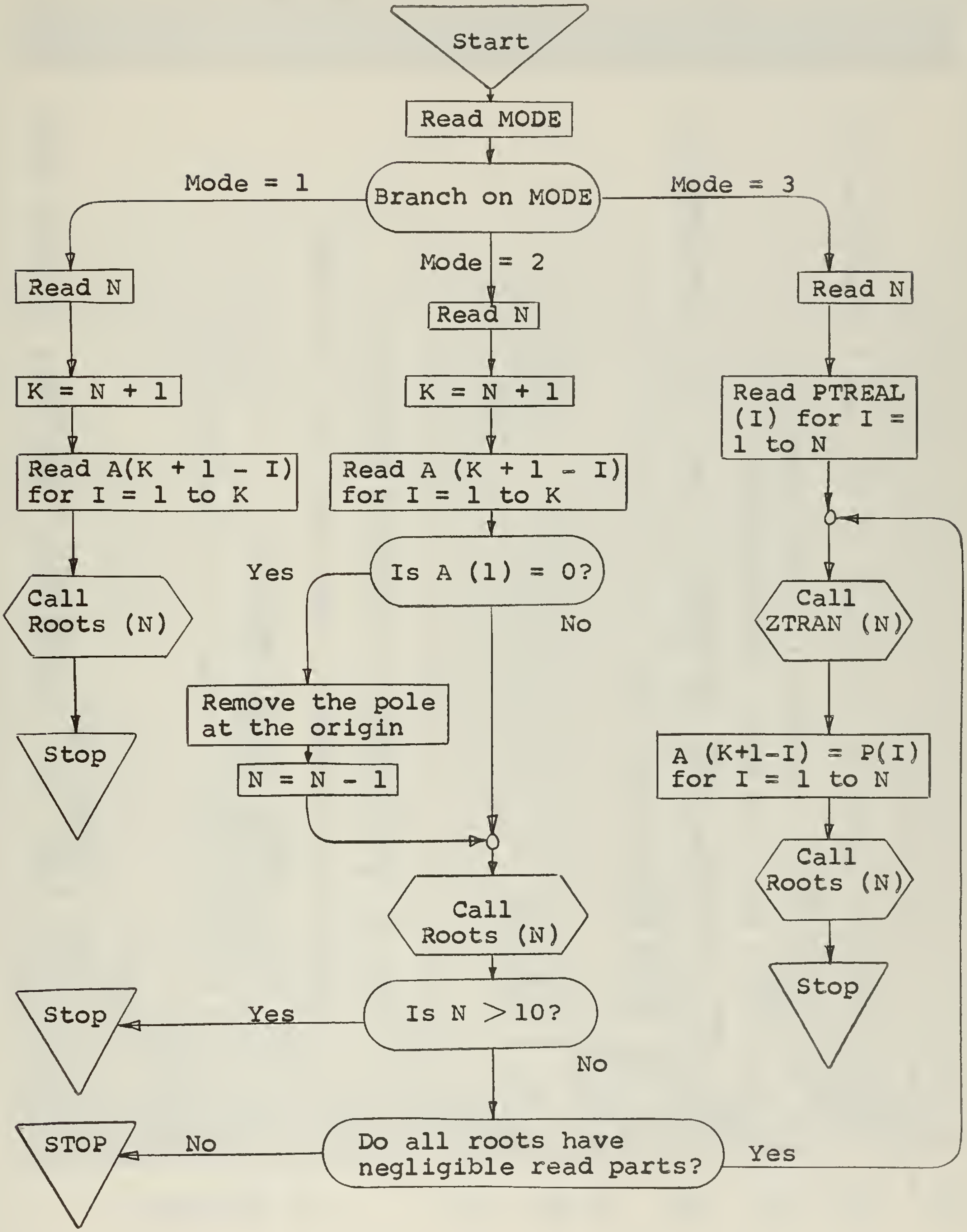
The data cards are prepared as shown below:

Column 1	Column 11	Column 21	Column 31	Column 41	Column 51	Column 61	Column 71	
.05	1606423500.0							Card 4

Column 1	Column 11	Column 21	Column 31	Column 41	Column 51	Column 61	Column 71	
-11.0	-17.0	-23.0	-83.0					Card 3
04								Card 2
3								Card 1

The answers are the same as those obtained in the example problem for Mode 2.

FLOW DIAGRAM FOR PROGRAM STAPP1



```
      PROGRAM STAPP1
C     COMBINATION ROOTFINDER AND Z-TRANSFORMER.  AVAILABLE IN THREE MODES
      DIMENSION A(30), PTREAL(30), PTIMAG(30), EXP(30)
      COMMON A, PTREAL, PTIMAG, EXP
      READ 9, MODE
      PRINT 12, MODE
    9 FORMAT(I1)
   10 FORMAT(I2)
   11 FORMAT((8F10.0))
   12 FORMAT(35H PROGRAM STAPP1   OPERATING IN MODE,I2,///)
   13 FORMAT(16H AS A ROOTFINDER)
   14 FORMAT(/////,18H FACTORING P(Z) BY)
   15 FORMAT(26H INITIALLY AS A ROOTFINDER)
      GO TO (20,30,50), MODE
   20 READ 10, N
C     N IS THE ORDER OF THE POLYNOMIAL WHOSE ROOTS ARE TO BE FOUND
      K=N+1
      READ 11, (A(K+1-I )  I=1,K)
C     A(I) IS ARRAY OF COEFFICIENTS IN DESCENDING POWERS OF Z
      PRINT 13
      CALL ROOTS(N)
      STOP
   30 READ 10, N
      K=N+1
      READ 11, (A(K+1-I)   I=1,K)
      IF(A(1)) 33,31,33
   31 DO 32  I=1,N
   32 A(I) = A(I+1)
      N = N-1
   33 PRINT 15
      CALL ROOTS(N)
C     TEST FOR N NOT GREATER THAN TEN
      IF(10-N) 34,36,36
   34 PRINT 35, N
   35 FORMAT(//////,85H SORRY, WE ARE UNABLE TO TAKE THE Z-TRANSFORM FOR
     1 MORE THAN TEN ROOTS ...AND YOU HAVE, I3)
      STOP
C     CHECKING ROOTS FOR NEGLIGIBLE IMAGINARY PARTS
   36 DO 39   I=1,N
      DIFF = (PTREAL(I))**2 - 10000.0*(PTIMAG(I))**2
      IF(DIFF) 37,37,39
   37 PRINT 38
   38 FORMAT(//////,63H SORRY, WE ARE UNABLE TO TAKE THE Z-TRANSFORM FOR
     1 COMPLEX ROOTS)
      STOP
   39 CONTINUE
      GO TO 51
   50 READ 10, N
```

[illegible]

```
      READ 11, (PTREAL(I)  I=1,N)
51    CALL ZTRAN(N)
      K = N+1
      DO 53   I=1,K
53    EXP(I) = A(I)
      DO 55  I = 1,K
55    A(K+1-I) = EXP(I)
      PRINT 14
      CALL ROOTS(N)
      STOP
      END
      SUBROUTINE ZTRAN(N)
      DIMENSION P(30), Q(30), EXP(30), COEFZ(30), FEXP(30), COZ(30),
     1          SYN(30), A(30), C(30), COEF(30)
      COMMON P, EXP, C, Q
      READ 100, PERIOD, VK
100   FORMAT(F10.0, F20.0)
      PRINT 101, PERIOD
101   FORMAT(/////,49H FINDING THE Z-TRANSFORM FOR A SAMPLING PERIOD OF,
     1     F6.3, 8H SECONDS,/)
      PRINT 102,N
102   FORMAT(20H NO. OF REAL POLES =,I2,/)
      PRINT 103, VK
103   FORMAT(17H PLANT GAIN, VK =, F20.3,/)
      PRINT 104,(I, EXP(I), I=1,N)
104   FORMAT(12H POLE NUMBER,I2,2H =,F20.8)
      DO 110   I=1,30
      C(I) = 0.0
      P(I) = 0.0
110   Q(I) = 0.0
      KSWITCH = 0
480   KSWITCH = KSWITCH + 1
      KK = 0
490   KK = KK + 1
      COEF(KK) = 0.0
      J1 = 0
500   J1 = J1 + 1
      IF(N-KK+1 - J1) 700,502,502
502   GO TO (600,505,505,505,505,505,505,505,505,505), KK
505   J2 = J1
510   J2 = J2 +1
      IF(N-KK+2 - J2) 500,512, 512
512   GO TO (600,600,515,515,515,515,515,515,515,515), KK
515   J3 = J2
520   J3 = J3 + 1
      IF(N-KK+3 - J3) 510,522,522
522   GO TO (600,600,600,525,525,525,525,525,525,525), KK
525   J4 = J3
530   J4 = J4 + 1
```

```
      IF(N-KK+4 - J4) 520,532,532
  532 GO TO (600,600,600,600,535,535,535,535,535,535), KK
  535 J5 = J4
  540 J4 = J4 + 1
      IF(N-KK+5 - J5)530,542,542
  542 GO TO (600,600,600,600,500,545,545,545,545,545), KK
  545 J6 = J5
  550 J6 = J6 + 1
      IF(N-KK+6 - J6)540,552,552
  552 GO TO (600,600,600,600,500,600,555,555,555,555), KK
  555 J7 = J6
  560 J7 = J7 + 1
      IF(N-KK+7 - J7)550,562,562
  562 GO TO (600,600,600,600,500,600,600,565,565,565), KK
  565 J8 = J7
  570 J8 = J8 + 1
      IF(N-KK+8 - J8)560,572,572
  572 GO TO (600,600,600,600,500,600,600,600,575,575), KK
  575 J9 = J8
  580 J9 = J9 + 1
      IF(N-KK+9 - J9)570,582,582
  582 GO TO (600,600,600,600,500,600,600,600,600,585), KK
  585 J10= J9
  590 J10= J10 +1
      IF(N-KK+10-J10) 580,600,600
  600 PEXP = 1.0
      FEXP(1) = EXP(J1)
      FEXP(2) = EXP(J2)
      FEXP(3) = EXP(J3)
      FEXP(4) = EXP(J4)
      FEXP(5) = EXP(J5)
      FEXP(6) = EXP(J6)
      FEXP(7) = EXP(J7)
      FEXP(8) = EXP(J8)
      FEXP(9) = EXP(J9)
      FEXP(10)= EXP(J10)
      DO 609 I=1,KK
  609 PEXP = PEXP * (-FEXP(I))
  610 COEF(KK) = COEF(KK) + PEXP
  630 GO TO (500,510,520,530,540,550,560,570,580,590), KK
  700 IF(N - KK)710,710,490
  710 CONTINUE
      GO TO (190,715), KSWITCH
C     COMPUTE DENOMINATOR COEFFICIENTS OF H(S)
  190 K = N +1
      A(K+1) = 1.0
      DO 200 I=1,N
  200 A(K+1-I) = COEF(I)
      A(1) = 0.0
```

```
      K = K + 1
      PRINT 205
 205  FORMAT(/,54H ASCENDING COEFFICIENTS OF PLANT DIFFERENTIAL EQUATION
     1)
      PRINT 206, ( I, A(I),  I=1,K)
 206  FORMAT(10X,2HA(,I2,2H)=,F20.6)
C     COMPUTE PARTIAL FRACTION COEFFICIENTS
      AC = 1.0/COEF(N)
C     COMPUTE SYNTHETIC DIVISION COEFFICIENTS
      DO 240  J=1,N
      SYN(1) = 1.0
      DO 220 I = 2,N
 220  SYN(I) = COEF(I-1) + EXP(J)*SYN(I-1)
C     EVALUATE DENOM
      DENOM = 0.0
      S =  EXP(J)
      DO 230  I=1,N
 230  DENOM = DENOM + SYN(I) * S ** (N-I)
      DENOM = DENOM * S ** 2
 240  C(J) = 1.0/DENOM
      BC = 0.0
      DO 250 I=1,N
 250  BC = BC - C(I)
      PRINT 255
 255  FORMAT(/,43H PARTIAL FRACTION COEFFICIENTS FOR UNITY VK)
      PRINT 256, AC, BC, (I, C(I),  I=1,N)
 256  FORMAT(10X,2HA=,E20.8,/, 10X,2HB=,E20.8,/,(6X,2HC(,I2,2H)=,E20.8))
      DO 260 I=1,N
 260  EXP(I) = 2.71828183 ** ( EXP(I) * PERIOD)
      PRINT 270
 270  FORMAT(/,33H LOCATION OF POLES IN THE Z-PLANE)
      PRINT 275,(I, EXP(I), I=1,N)
 275  FORMAT(3X,17H Z-PLANE POLE NO.,I2,2H =,E16.8)
      GO TO 480
C     CONTRIBUTION TO P(Z) FROM TERM A OF P. F. EXPANSION
 715  P(1) = AC*PERIOD
      DO 720  I = 1,N
 720  P(I+1) = AC*PERIOD*COEF(I)
C     COMPUTATION OF Q(Z)
      Q(1) = 1.0
      Q(2) = COEF(1) - 1.0
      DO 740 I=2,N
 740  Q(I+1) = COEF(I) - COEF(I-1)
      Q(N+2) = -COEF(N)
C     CONTRIBUTION TO P(Z) FROM TERM (B)
      K = N + 1
      DO 750  I=1,K
 750  P(I) = P(I) + BC * Q(I+1)
C     COMPUTE Q(Z)(Z-1)
```

```
      COZ(1) = 1.0
      COZ(2) = Q(2) - 1.0
      DO 760  I=1,N
 760  COZ(I+2) = Q(I+2) - Q(I+1)
      COZ(N+3) = -Q(N+2)
C     CONTRIBUTIONS FROM C(I) TERMS
      DO 790  J=1,N
      SYN(1) = COZ(2) + COZ(1)*EXP(J)
      DO 770  I=1,N
 770  SYN(I+1) = COZ(I+2) + SYN(I)*EXP(J)
      DO 775  I=1,K
 775  P(I) = P(I) + C(J)*SYN(I)
 790  CONTINUE
      DO 795  I=1,K
 795  P(I) = P(I)*VK
      PRINT 800
 800  FORMAT(///,6X,21H COEFFICIENTS OF P(Z))
      PRINT 805,(I, P(I), I=1, 15)
 805  FORMAT(3H P(,I2,2H)=,E20.8)
      PRINT 806
 806  FORMAT(/,6X,21H COEFFICIENTS OF Q(Z))
      PRINT 810,(I, Q(I), I=1, 15)
 810  FORMAT(3H Q(,I2,2H)=,E20.8)
      END
      SUBROUTINE ROOTS(N)
      DIMENSION A(30), PTREAL(30), PTIMAG(30), Q(30), T(30), QR(30)
      COMMON A, PTREAL, PTIMAG, Q
      DO 812   I=1,30
      Q(I) = 0.0
      T(I) = 0.0
 812  QR(I)= 0.0
      J = 1
      K = N+1
      IF(A(3)*A(4)) 815,814,815
 814  PP = 0.000001
      QQ = PP
      GO TO 816
 815  PP = 0.0
      QQ = 0.0
 816  CONTINUE
      ITERATE = 0
      PRINT 817, N, (I, A(I), I=1,K)
 817  FORMAT(50H FINDING THE ROOTS OF F(Z) = A(N+1)Z**N +...+ A(1), //,
     1       9X,21H FOR THIS PROBLEM, N=, I2, 9H  AND THE, //,
     2       9X,35H COEFFICIENTS OF THE POLYNOMIAL ARE, //,
     3      (10X, 2HA(, I2, 2H)=, E20.8),    /////)
      IF(N-2) 818,820,823
 818  Q(3) = A(1)
      Q(4) = A(2)
```

```
      GO TO 865
  820 Q(3) = A(1)
      Q(4) = A(2)
      Q(5) = A(3)
      GO TO 870
  823 CONTINUE
      DPPMAX = 0.0000000001
      DQQMAX=DPPMAX
      DO 825 I=1,K
  825 QR(I) = A(I)
      NREM = N
      PRINT 827, NREM
  827 FORMAT(////,37H ITERATION FOR A POLYNOMIAL OF DEGREE, I3, /)
  828 K=NREM+1
C     RESTORE ORIGINAL POLYNOMIAL FOR THIS ITERATION
      DO 829   I=1,K
  829 Q(I) = QR(I)
      ITERATE = ITERATE + 1
C     DIVISION OF POLYNOMIAL Q(Z) BY TRIAL QUADRATIC Z**2-PP*Z-QQ
      DO 830 I=1,K
  830 Q(NREM+2-I) = Q(NREM+2-I) + PP*Q(NREM+3-I) + QQ*Q(NREM+4-I)
      K = NREM-1
C     DIVISION OF REDUCED Q(Z) TO FORM POLYNOMIAL T(Z)
      DO 832 I=1,K
  832 T(NREM-I)=Q(NREM+2-I)+PP*T(NREM+1-I)+QQ*T(NREM+2-I)
C     COMPUTATION OF CORRECTIONS TO PP AND QQ
      FACTM = PP*T(1) + QQ*T(2)
      DENOM = (T(1))**2 - FACTM*T(2)
      DPP=(T(2)*Q(1)-T(1)*Q(2))/DENOM
      DQQ=(FACTM*Q(2)-T(1)*Q(1))/DENOM
      PRINT 836, ITERATE,PP,DPP,QQ,DQQ
  836 FORMAT(14H ITERATION NO.,I3,5X,3HPP=,E17.9,5X,4HDPP=,
     1 E12.4,5X,3HQQ=,E17.9,5X,4HDQQ=,E12.4)
     1        F20.8)
      IF(DPP*DQQ) 840,850,840
  840 IF(DPPMAX-DPP) 848,848,841
  841 IF(DPPMAX+DPP) 848,848,842
  842 IF(DQQMAX-DQQ) 848,848,843
  843 IF(DQQMAX+DQQ) 848,848,850
C     CORRECTING PP AND QQ FOR NEXT ITERATION
  848 PP=PP+DPP
      QQ=QQ+DQQ
      DPPMAX = (PP/100000.0)**2
      DQQMAX = (QQ/100000.0)**2
      IF(50 - ITERATE) 849,849,828
  849 PRINT 847
  847 FORMAT(///,58H TRIAL DIVISOR ASSUMED SATISFACTORY AFTER FIFTY ITER
     1ATIONS)
C     REMAINDER TERMS ARE NEGLIGIBLE, SO FIND ROOTS OF QUADRATIC DIVISOR
```

```
 850  PTREAL(J)=PP/2.0
      DESCRIM=(PP/2.0)**2 + QQ
      IF(DESCRIM) 852,855,855
C     ROOTS ARE COMPLEX
 852  DESCRIM = -DESCRIM
      PTIMAG(J)=SQRTF(DESCRIM)
      PTREAL(J+1)=PTREAL(J)
      PTIMAG(J+1)= -PTIMAG(J)
      GO TO 860
C     ROOTS ARE REAL
 855  PART2=SQRTF(DESCRIM)
      PTREAL(J+1) = PTREAL(J) - PART2
      PTIMAG(J+1) = 0.0
      PTREAL(J) = PTREAL(J) + PART2
      PTIMAG(J) = 0.0
 860  PRINT 861, PTREAL(J), PTIMAG(J)
      PRINT 861, PTREAL(J+1), PTIMAG(J+1)
 861  FORMAT(12H REAL PART =,F20.14,5X,16HIMAGINARY PART =,F20.14)
      NREM=NREM-2
      IF(NREM-2) 865,870,880
C     REDUCED POLYNOMIAL IS OF THE FIRST ORDER
 865  PTREAL(N)= -Q(3)/Q(4)
      PTIMAG(N) = 0.0
      GAIN = Q(4)
      GO TO 900
C     REDUCED POLYNOMIAL IS A QUADRATIC
 870  PTREAL(N-1)= -Q(4)/(2.0*Q(5))
      DESCRIM = (Q(4))**2 - 4.0*Q(5)*Q(3)
      IF(DESCRIM) 872,875,875
 872  DESCRIM = -DESCRIM
      PTIMAG(N-1) = (SQRTF(DESCRIM))/(2.0*Q(5))
      PTREAL(N) = PTREAL(N-1)
      PTIMAG(N) = -PTIMAG(N-1)
      GAIN = Q(5)
      GO TO 900
 875  PART2 = (SQRTF(DESCRIM))/(2.0*Q(5))
      PTREAL(N) = PTREAL(N-1) - PART2
      PTIMAG(N) = 0.0
      PTREAL(N-1) = PTREAL(N-1) + PART2
      PTIMAG(N-1) = 0.0
      GAIN = Q(5)
      GO TO 900
C     REDUCED POLYNOMIAL IS A CUBIC OR BETTER.  FORM IT FOR NEXT DIVISION
 880  K=NREM+1
      DO 885 I=1,K
 885  QR(I) = Q(I+2)
      J=J+2
      DO 886   I=1,2
      Q(K+I) = 0.0
```

```
      QR(K+I) = 0.0
  886 T(NREM-1+I) = 0.0
      PRINT 827, NREM
      NN=N+1
      PRINT 890, (I,  QR(I),    I=1,NN)
  890 FORMAT(23H WHOSE COEFFICIENTS ARE,//,(9X,3HQR(,I2,2H)=,F20.8),//)
      ITERATE = 0
      IF(QR(3)*QR(4)) 895,828,895
C     USE PREVIOUS TRIAL DIVISOR IN CASE OF MULTIPLE ROOTS.  OTHERWISE
C     START FROM ORIGIN AGAIN TO APPROACH NEXT ROOT OF REDUCED POLYNOMIAL
  895 PP = 0.0
      QQ = 0.0
      GO TO 828
  900 PRINT 920, GAIN, (I, PTREAL(I), PTIMAG(I), I=1,N)
  920 FORMAT(///,27H THE CONSTANT MULTIPLIER IS, F20.8,//,
     1         32H THE ROOTS OF THE POLYNOMIAL ARE,//,
     2         18X, 11H(REAL PART),14X, 16H(IMAGINARY PART), /,
     3        (6H ROOT(, I2, 2H)=, E20.8, 5X, E20.8))
      SUM = 0.0
      DO 930 I=1,N
  930 SUM = SUM + PTREAL(I)
      ACTUAL =-A(N)/A(N+1)
      PRINT 935, SUM,ACTUAL
  935 FORMAT(/,42H AS A CHECK, THE SUM OF THE ABOVE ROOTS IS, F25.9,/,
     1          42H WHEN, IN ACTUAL FACT, THE SUM SHOULD BE  , F25.9)
      END
      END
```

APPENDIX III

Determination of stability of a third-order system as sampling rate is decreased

$$G(z) = K\left(\frac{z-1}{z}\right) \mathcal{Z}\left[\frac{A}{s^2} + \frac{B}{s} + \frac{C}{s+25} + \frac{D}{s+70}\right]$$

where $K = 80{,}000$

$A = 57.1428 \times 10^{-5}$

$B = -3.1020 \times 10^{-5}$

$C = 3.555 \times 10^{-5}$

$D = -0.4535 \times 10^{-5}$

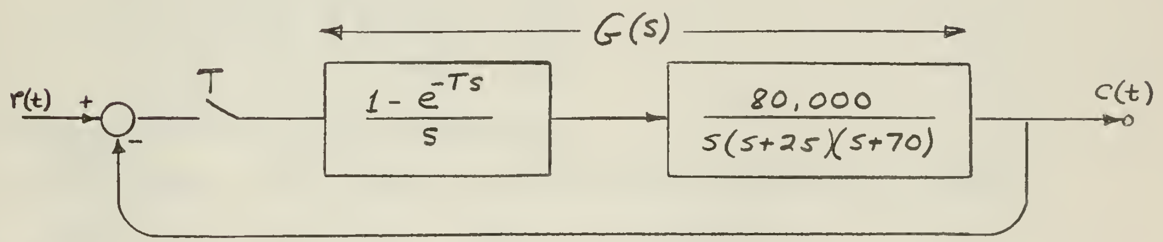
or

$$\frac{P(z)}{Q(z)} = \frac{(z-1)}{z} K \left[\frac{ATz}{(z-1)^2} + \frac{Bz}{z-1} + \frac{Cz}{z-e^{-25T}} + \frac{Dz}{z-e^{-70T}}\right]$$

Multiplying out:

$$Q(z) = z^3 - \left[e^{-25T} + e^{-70T} + 1\right] z^2 + \left[e^{-25T} + e^{-70T} + e^{-95T}\right] z - e^{-95T}$$

$$\begin{aligned} P(z) = {} & K\left[AT - B(e^{-25T} + e^{-70T} + 1) - C(2 + e^{-70T}) - D(2 + e^{-25T})\right] z^2 \\ & + K\left[-AT(e^{-25T} + e^{-70T}) + B(e^{-25T} + e^{-70T} + e^{-95T}) + C(1 + 2e^{-70T}) + D(1 + 2e^{-25T})\right] z \\ & + K\left[ATe^{-95T} - Be^{-95T} - Ce^{-70T} - De^{-25T}\right] \end{aligned}$$



Characteristic equation: $A(z) = P(z) + Q(z) = a_3 z^3 + a_2 z^2 + a_1 z + a_0$

where $a_3 = 1$

$$a_2 = KAT - KBe^{-25T} - KBe^{-70T} - 2KC - KCe^{-70T} - 2KD - KDe^{-25T} - KB - e^{-25T} - e^{-70T} - 1$$

$$\begin{aligned} a_1 = {} & -KATe^{-25T} - KATe^{-70T} + KBe^{-25T} + KBe^{-70T} + KBe^{-95T} + KC + 2KCe^{-70T} \\ & + KD + 2KDe^{-25T} + e^{-25T} + e^{-70T} + e^{-95T} \end{aligned}$$

$$a_0 = KATe^{-95T} - KBe^{-95T} - KCe^{-70T} - KDe^{-25T} - e^{-95T}$$

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

Note that G(z) has been expressed in positive powers of z since we anticipate making a change of variables. Later on, z-transforms are expressed using inverse powers of z in order to correspond to the concept of difference equations.

Using the Bilinear Transformation: $z = \frac{1+\omega}{1-\omega}$

$$A(z) = a_3\left(\frac{1+\omega}{1-\omega}\right)^3 + a_2\left(\frac{1+\omega}{1-\omega}\right)^2 + a_1\left(\frac{1+\omega}{1-\omega}\right) + a_0$$

$$= a_3\left[1+3\omega+3\omega^2+\omega^3\right] + a_2\left[1+\omega-\omega^2+\omega^3\right] + a_1\left[1-\omega-\omega^2+\omega^3\right] + a_0\left[1-3\omega+3\omega^2-\omega^3\right]$$

$$= \omega^3\left[a_3-a_2+a_1-a_0\right] + \omega^2\left[3a_3-a_2-a_1+3a_0\right] + \omega\left[3a_3+a_2-a_1-3a_0\right] + \left[a_3+a_2+a_1+a_0\right]$$

Routh-Hurwitz criterion for third order equation:

all roots in left half plane if $b_1 b_2 - b_0 b_3 > 0$

where
$$b_0 = a_3 + a_2 + a_1 + a_0$$
$$b_1 = 3a_3 + a_2 - a_1 - 3a_0$$
$$b_2 = 3a_3 - a_2 - a_1 + 3a_0$$
$$b_3 = a_3 - a_2 + a_1 - a_0$$

In terms of the system parameters,

$$b_0 = e^{-95T}(KAT) + e^{-70T}(-KAT) + e^{-25T}(-KAT) + (KAT - KC - KD)$$

$$b_1 = e^{-95T}(-3KAT + 2KB + 2) + e^{-70T}(KAT - 2KB - 2) + e^{-25T}(KAT - 2KB - 2) + (KAT - 3KC - 3KD + 2)$$

$$b_2 = e^{-95T}(3KAT - 4KB - 4) + e^{-70T}(KAT - 4KC) + e^{-25T}(KAT - 4KD) + (-KAT + KC + KD + 1)$$

$$b_3 = e^{-95T}(KAT) + e^{-70T}(-KAT + 2KB + 2KC + 2) + e^{-25T}(-KAT + 2KB + 2KD + 2) + (-KAT + 3KC + 3KD + 1)$$

A computer program to solve the above equations for one hundred values of T and the results obtained are shown on the following pages.

```
..JOB STAPP2
      PROGRAM STAPP2
C     DETERMINATION OF ROUTH-HURWITZ CRITERION FOR THIRD
      ORDER SYSTEM
C     AS SAMPLING PERIOD IS VARIED
      VK=80000.
      A= .000571428
      B=-.000031020
      C= .000035555
      D=-.000004535
      T=0.0
      N=0
      PRINT 199
  199 FORMAT(32H     PERIOD       ROUTH CRITERION)
  100 T=T+.001
      EXP25=2.71828**(-25.0*T)
      EXP70=2.71828**(-70.0*T)
      EXP95=2.71828**(-95.0*T)
      A0=EXP95*(VK*A*T-VK*B-1.)-EXP70*VK*C-EXP25*VK*D
     0A1=EXP95*(VK*B+1.)+EXP70*(-VK*A*T+VK*B+2.*VK*C+1.)+
     1   EXP25*(-VK*A*T+VK*B+2.0*VK*D+1.)+VK*C+VK+D
     0A2=EXP70*(-VK*B-VK*C-1.)+EXP25*(-VK*B-VK*D-1.)+VK*
      (A*T-2.*C-2.*D)
     1   -1.0-VK*B
      A3=1.0
      B0=A3+A2+A1=A0
      B1=3.0*A3+A2-A1-3.0*A0
      B2=3.0*A3-A2-A1+3.0*A0
      B3=A3-A2+A1-A0
      ROUTH= B1*B2-B0*B3
      PRINT 200,T,ROUTH
  200 FORMAT(F10.4, F20.5)
      N=N+1
      IF(N-100)100,300,300
  300 STOP
      END
      END
```

(FORTRAN program for the CDC 1604)

PERIOD	ROUTH CRITERION	PERIOD	ROUTH CRITERION
.0010	.00060	.0510	-2.33345
.0020	.00418	.0520	-2.44305
.0030	.01226	.0530	-2.55283
.0040	.02524	.0540	-2.66273
.0050	.04279	.0550	-2.77266
.0060	.06411	.0560	-2.88256
.0070	.08814	.0570	-2.99235
.0080	.11372	.0580	-3.10199
.0090	.13966	.0590	-3.21141
.0100	.16483	.0600	-3.32057
.0110	.18818	.0610	-3.42940
.0120	.20879	.0620	-3.53787
.0130	.22587	.0630	-3.64593
.0140	.23874	.0640	-3.75354
.0150	.24688	.0650	-3.86066
.0160	.24986	.0660	-3.96725
.0170	.24736	.0670	-4.07329
.0180	.23919	.0680	-4.17874
.0190	.22522	.0690	-4.28358
.0200	.20541	.0700	-4.38776
.0210	.17977	.0710	-4.49128
.0220	.14839	.0720	-4.59410
.0230	.11137	.0730	-4.69620
.0240	.06889	.0740	-4.79756
→ .0250	STABILITY .02112	.0750	-4.89816
.0260	LIMIT -.03171	.0760	-4.99798
.0270	-.08939	.0770	-5.09700
.0280	-.15167	.0780	-5.19522
.0290	-.21831	.0790	-5.29261
.0300	-.28905	.0800	-5.38915
.0310	-.36365	.0810	-5.48485
.0320	-.44185	.0820	-5.57967
.0330	-.52341	.0830	-5.67363
.0340	-.60809	.0840	-5.76669
.0350	-.69565	.0850	-5.85886
.0360	-.78588	.0860	-5.95012
.0370	-.87855	.0870	-6.04048
.0380	-.97345	.0880	-6.12991
.0390	-1.07039	.0890	-6.21842
.0400	-1.16919	.0900	-6.30599
.0410	-1.26965	.0910	-6.39264
.0420	-1.37162	.0920	-6.47834
.0430	-1.47493	.0930	-6.56309
.0440	-1.57944	.0940	-6.64690
.0450	-1.68499	.0950	-6.72977
.0460	-1.79146	.0960	-6.81168
.0470	-1.89873	.0970	-6.89263
.0480	-2.00666	.0980	-6.97264
.0490	-2.11516	.0990	-7.05169
.0500	-2.22412	.1000	-7.12979

Determination of Routh-Hurwitz criterion for a third order system as sampling period, T, goes from .001 to 0.1 seconds.